

Rowan University

## Rowan Digital Works

---

Theses and Dissertations

---

4-26-2018

### Risk analysis of autonomous vehicle and its safety impact on mixed traffic stream

Plaban Das  
*Rowan University*

Follow this and additional works at: <https://rdw.rowan.edu/etd>



Part of the [Transportation Engineering Commons](#)

**Let us know how access to this document benefits you - share your thoughts on our feedback form.**

---

#### Recommended Citation

Das, Plaban, "Risk analysis of autonomous vehicle and its safety impact on mixed traffic stream" (2018). *Theses and Dissertations*. 2545.  
<https://rdw.rowan.edu/etd/2545>

This Thesis is brought to you for free and open access by Rowan Digital Works. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Rowan Digital Works. For more information, please contact [LibraryTheses@rowan.edu](mailto:LibraryTheses@rowan.edu).

**RISK ANALYSIS OF AUTONOMOUS VEHICLE AND ITS SAFETY IMPACT  
ON MIXED TRAFFIC STREAM**

by  
Plaban Das

A Thesis

Submitted to the  
Department of Civil and Environmental Engineering  
College of Engineering  
In partial fulfillment of the requirement  
For the degree of  
Master of Science in Civil Engineering  
at  
Rowan University  
December 22, 2017

Thesis Chair: Dr. Parth Bhavsar



## **Dedications**

This thesis is dedicated to my parents,

Mrs. Archana Das and Mr. Santi Ranjan Das.

## **Acknowledgments**

I still remember my first day at Rowan University; it was a sunny, cold morning. I stepped in the engineering building with joy and a dream. Today I am standing in front of that same building accomplishing one of my lifelong dreams. From that first day to today, I have gathered lots of memories at this university, memories of successes and also memories of failures at the same time. The moments of successes inspired me to chase a new dream and led me to today's achievement. On the other hand, the moments of failure made me stronger and stimulated my determination to achieve this degree. I would like to thank Lord Krishna, the Almighty, for showering His blessing upon me throughout my life to achieve all these successes and also for showing me the right path in all difficulties.

From bottom of my heart, I would like to express my deepest gratitude to my advisor, Dr. Parth Bhavsar, Assistant Professor, Department of Civil and Environmental Engineering, Rowan University, for giving me the opportunity to join his research lab and providing his constant motivation and encouragement throughout my pursuit of the M.S. degree. His dynamic thinking and deep knowledge in the transportation field always inspired me to conduct thorough research for my thesis. Without his constructive comments and guidance, it would not have been possible for me to complete this thesis successfully. It was a great privilege and honor for me to conduct research under his supervision. I would like to thank Dr. Mashrur Chowdhury, Professor, Glenn Department of Civil Engineering, Clemson University, for his invaluable suggestions and guidelines throughout this thesis work which have led to the completion of this thesis. Special thanks to Dr. Yusuf Mehta, Professor, Department of Civil and Environmental

Engineering, Rowan University, for spending his valuable time reading my thesis and providing helpful suggestions.

I want to cordially thank Dr. Kakan Dey, Assistant Professor, Department of Civil and Environmental Engineering, West Virginia University, for his suggestions during my work on this thesis. My special thanks to Dr. Karzan Bahaaldin for his support in simulation modeling. Many thanks to my clinic students, including Matthew Paugh, Joel Levin, Robert Reiss, Brendan Rahl, and David Schornstaedt for their support during this research. I also extend sincere thanks to Richard Farally for his help regarding software installation.

I am extremely grateful to my beloved mother, Archana Das and father, Santi Ranjan Das, for their love, nurture, prayers and sacrifices—for educating and inspiring me to pursue this degree. Special thanks to my little sister, Doctor Tule Das, for encouraging me at all times. I want to cordially thank Engr. Kamrul Ahsan and his family for providing me moral support always and their delicious food over the years. Sincere thanks go to Dr. Shyamal Kumar Das and his family for providing me guidance and support regarding my admission to the university. Many thanks go to Dr. Tariq Ahmed and his family for letting me stay in their house for a week before finding an accommodation. I also want to thank my friends, Godfrey, Abraham, Harsh, Iftekhhar, Sami, Bijoy, Shantu and Liton for spending their time with me as friends. Finally, I would like to say a big thank you to my wife, Asha, for her love, cooperation and patience throughout the duration of this research.

## **Abstract**

Plaban Das

### **RISK ANALYSIS OF AUTONOMOUS VEHICLE AND ITS SAFETY IMPACT ON MIXED TRAFFIC STREAM**

2017-2018

Dr. Parth Bhavsar

Master of Science in Civil Engineering

In 2016, more than 35,000 people died in traffic crashes, and human error was the reason for 94% of these deaths. Researchers and automobile companies are testing autonomous vehicles in mixed traffic streams to eliminate human error by removing the human driver behind the steering wheel. However, recent autonomous vehicle crashes while testing indicate the necessity for a more thorough risk analysis. The objectives of this study were (1) to perform a risk analysis of autonomous vehicles and (2) to evaluate the safety impact of these vehicles in a mixed traffic stream. The overall research was divided into two phases: (1) risk analysis and (2) simulation of autonomous vehicles. Risk analysis of autonomous vehicles was conducted using the fault tree method. Based on failure probabilities of system components, two fault tree models were developed and combined to predict overall system reliability. It was found that an autonomous vehicle system could fail 158 times per one-million miles of travel due to either malfunction in vehicular components or disruption from infrastructure components. The second phase of this research was the simulation of an autonomous vehicle, where change in crash frequency after autonomous vehicle deployment in a mixed traffic stream was assessed. It was found that average travel time could be reduced by about 50%, and 74% of conflicts, i.e., traffic crashes, could be avoided by replacing 90% of the human drivers with autonomous vehicles.

## Table of Contents

Abstract .....	vi
List of Figures .....	xi
List of Tables .....	xiv
Chapter 1: Introduction .....	1
Background and Motivation .....	5
Research Objectives.....	9
Organization of Thesis .....	10
Chapter 2: Literature Review .....	12
Autonomous Vehicle .....	13
Level of Automation .....	15
System Disintegration .....	17
Vehicular Sensors for Automation.....	25
Risk Analysis .....	31
Phases of Risk Analysis .....	31
Elements of Risk Analysis .....	32
Classification of Risk Analysis Techniques .....	34
Risk Analysis of Autonomous Vehicles .....	35
Situation-Based Risk Analysis Method .....	36
Ontology-Based Risk Analysis Method.....	39
Fault Tree-Based Risk Analysis Method .....	41
Fault Tree Analysis Structure .....	47
Fault Tree Mathematical Formulation .....	48



## Table of Contents (Continued)

Summary .....	49
Chapter 3: Method .....	50
Research Method .....	50
Risk Analysis .....	51
Risk Identification.....	52
Risk Estimation.....	52
Risk Hierarchization .....	53
Evaluation of Fault Tree Model.....	54
Online Survey .....	55
Developing Survey Instruments.....	55
Simulation of Autonomous Vehicle.....	58
Traffic Network Modeling.....	59
Number of Simulation Runs .....	60
Formulation of the Autonomous Navigation Algorithm .....	63
Modeling Multiple Scenarios.....	68
Conflict Analysis .....	69
Summary .....	74
Chapter 4: Risk Analysis of Autonomous Vehicle .....	75
Risk Identification.....	75
Autonomous Vehicle Components .....	76
Transportation Infrastructure Components .....	80
Risk Estimation.....	83

## **Table of Contents (Continued)**

Fault Tree for Autonomous Vehicular Component Failures .....	84
Fault Tree for Transportation Infrastructure Component Failures .....	85
Combined Fault Tree .....	87
Risk Hierarchization .....	90
Evaluation of Fault Tree Model .....	92
Summary .....	96
Chapter 5: Online Survey.....	97
Developing Survey Instruments.....	97
Survey Results .....	99
Analysis of Survey Results .....	100
Summary .....	102
Chapter 6: Autonomous Vehicle Simulation Results .....	103
Crash Frequency Estimation .....	103
Integration of Fault Tree and Simulation Modeling .....	116
Summary .....	118
Chapter 7: Conclusions and Recommendations .....	119
Recommendations.....	122
References.....	124
Appendix A: Calculation of Simulation Runs Number .....	149
Appendix B: External Driver Model Code (DLL File Development) .....	153
Appendix C: Code for Integration of Fault Tree and Simulation Modeling.....	170
Appendix D: Survey Calculation .....	176

## **Table of Contents (Continued)**

Appendix E: Travel Time Data for Travel Time Measurement Segment 1 .....	178
Appendix F: Conflict Analysis for Different Autonomous Vehicle Penetrations .....	180

## List of Figures

Figure	Page
Figure 1. Traffic fatalities per year in the United States .....	3
Figure 2. Infographic architecture of autonomous vehicle functional components .....	14
Figure 3. Comparison between NHTSA and SAE International levels of automation classification .....	17
Figure 4. Functional responsibilities of vehicular sensors.....	28
Figure 5. Classification of risk analysis methods: cost benefit analysis (CBA), fault tree analysis (FTA), stability analysis (SA), risk benefit analysis (RBA), Monte Carlo simulation (MCS), failure mode and effects analysis (FMEA), common mode common cause (CMCC), root cause analysis (RCA), risk compensation theory (RCT), and risk homeostasis theory (RHT) .....	35
Figure 6. An example of situation-based collision risks identification and evaluation (Hurst, 1996) .....	39
Figure 7. An example of ontology structure for autonomous vehicle risk analysis (Worrall et al., 2010).....	41
Figure 8. A sample graphical representation of fault tree analysis (Duran et al., 2013b) .....	44
Figure 9. An example of fault tree structure .....	47
Figure 10. Fault tree gates and events.....	48
Figure 11. Overall research methodology.....	51

## List of Figures (Continued)

Figure	Page
Figure 12. Steps for developing fault tree.....	53
Figure 13. Step by step methodology of risk analysis .....	54
Figure 14. Steps involved in online survey.....	57
Figure 15. Traffic simulation model of I-476 in Pennsylvania developed in Vissim .....	60
Figure 16. Flow of information between Vissim and EDM .....	66
Figure 17. External driver model algorithm .....	68
Figure 18. Three types of crash in SSAM (Pu & Joshi, 2008) .....	71
Figure 19. Integration platform of Vissim and SSAM .....	72
Figure 20. Integration platform of fault tree and traffic simulation model.....	74
Figure 21. Fault tree analysis considering failures due to vehicular components .....	86
Figure 22. Failures due to transportation infrastructure components .....	87
Figure 23. Failure of autonomous vehicles in mixed traffic streams using fault tree models .....	90
Figure 24. Comparison between the results of risk analysis and real-world incident percentages.....	95
Figure 25. Autonomous vehicle Delphi survey flow .....	98
Figure 26. Simulated travel time measurement segments (Source: Google Map)–not to scale.....	106
Figure 27. Average travel time over different random seed numbers .....	107

### List of Figures (Continued)

Figure	Page
Figure 28. Travel time reduction percentages over autonomous vehicle market shares for travel time measurement in segment 1 .....	108
Figure 29. Travel time reduction percentages over autonomous vehicle market shares for travel time measurement in segment 2 .....	109
Figure 30. Travel time reduction percentages over autonomous vehicle market shares for travel time measurement in segment 3 .....	110
Figure 31. Travel time reduction percentages over autonomous vehicle market shares for travel time measurement in segment 4 .....	110
Figure 32. Travel time reduction percentages over autonomous vehicle market shares for travel time measurement in segment 5 .....	111
Figure 33. Conflict reduction frequency with the increase of autonomous vehicle population in mainstream traffic mix .....	113
Figure 34. Variation of travel time between failure and non-failure scenarios .....	118

## List of Tables

Table	Page
Table 1. California DMV autonomous vehicle crash report.....	7
Table 2. Development of autonomous driving assistance technology.....	29
Table 3. Summary of risk analysis techniques used .....	45
Table 4. Failure probabilities of autonomous vehicular components .....	79
Table 5. Failure probabilities of basic transportation system infrastructure components .....	82
Table 6. Minimal cut-sets of autonomous vehicles components .....	92
Table 7. California DMV autonomous vehicles testing data.....	93
Table 8. Results of first round of survey.....	100
Table 9. Interpretation of Kendall's W .....	101
Table 10. Description of modeled travel time measurement segments .....	104
Table 11. Variations in conflict frequency reductions when TTC (= 0.9, 1.2 and 1.5) and when PET (= 4.0) .....	114
Table 12. Variations in conflict frequency reductions when TTC (= 0.9, 1.2 and 1.5) and when PET (= 3.0) .....	115

## **Chapter 1**

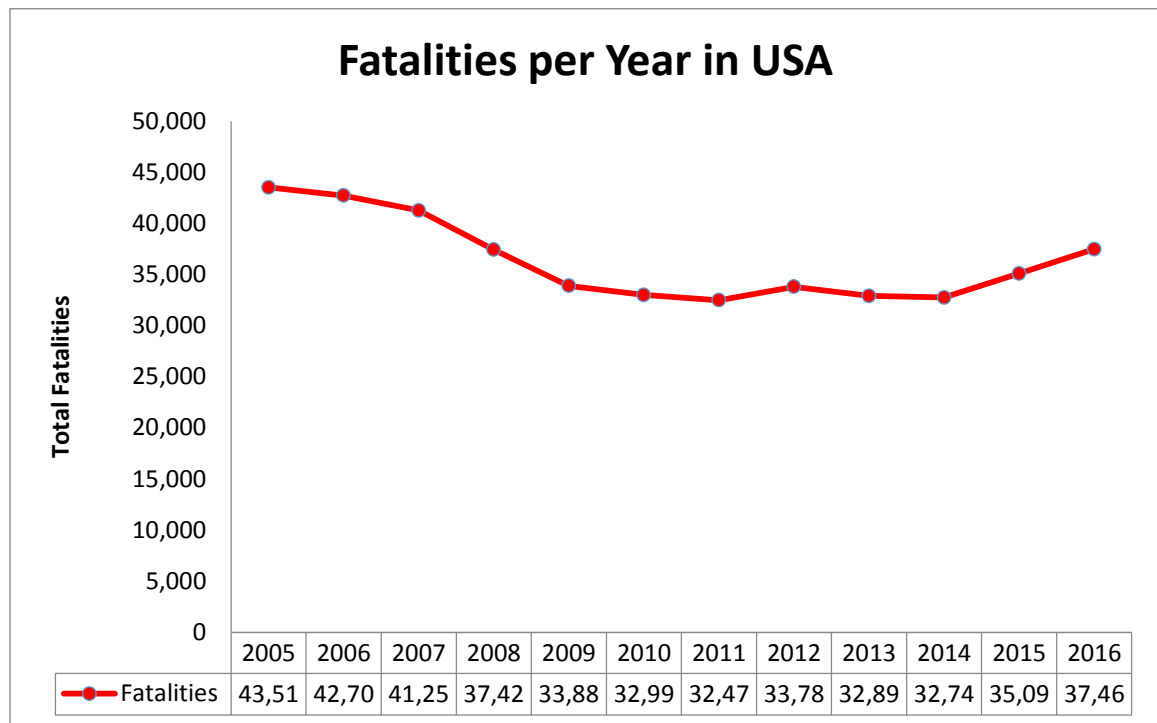
### **Introduction**

The transportation system, the key to sociological and economic progress, has ebbed and flowed throughout the history of mankind to reduce travel time and increase comfortability. After many trials and tribulations, we can now move ourselves and transfer goods from one place to another, nearby or distant, by selecting one or multiple transportation mode alternatives. Whether the mode is a personal vehicle or airplane, the design features are based on customer/user preferences and perspectives. The evolution of the transportation system has undergone many modifications, while many modes have become extinct. In the stone age of antiquity, humans walked and ran upon the solid earth with bare feet (Demartini, 2014). Afterwards they tamed horses and horses became the primary mode of transportation for many years. Even though horse-drawn vehicles were carrying 120,000 passengers per day in New York by the late 1860s; yet, they were unwelcome as they were driven at very slow speeds and had unpleasant byproducts (McShane & Tarr, 2007; Tarr., 1996). Moreover, 200 people were killed in New York by horses and horse-drawn vehicles in 1900 (McShane, 1995). However, transportation systems modernized after the industrial revolution. The innovation of steam engines in the late 18<sup>th</sup> century was the first major advancement for transportation. In 1801, the first steam engine automobile was exhibited in England. These first-generation automobiles were inefficient and had the same speed as horses. The automobile engine went through many further modifications over the next hundred years (Lab). Later, the combustion engine was invented, and automobiles became more efficient with faster speed. Moreover, personal transportation became more affordable day by day due to advances in



technologies. However, the overall transportation system has been overloaded due to the increased number of vehicles. According to a recent report card by the American Society of Civil Engineers (ASCE) (NHTSA, 2015), in 2014 traffic congestion cost \$160 billion in wasted time and fuel in the United States (U.S.) which averaged out to 42 hours per driver annually. More than 2 out of 5 miles of urban interstates are congested with high volume of traffic (NHTSA, 2015). It is not just valuable revenues and resources that are being wasted due the traffic congestion. Unfortunately, these congested conditions provoke road rage and risky driving behaviors (Salomon & Mokhtarian, 1997). Risky driving behavior leads to traffic crashes and results in morbidity (number of drivers with injuries that eventually lead to death) and mortality (actual accident death count). More lives have been lost in traffic crashes than from human diseases for last many years (Petridou & Moustaki, 2000). According to NHTSA traffic safety fact sheets, traffic crashes were responsible for more than 35,000 deaths on U.S. roadways (NHTSA, 2016a), and 10 out of the total 100 deaths caused by distracted driving (NHTSA, 2016d). It is important to include that human behavioral factors were responsible for 94% of total road crashes (Petridou & Moustaki, 2000). The traffic crash fatality trend in the U.S. per year from 2005 is presented in Figure 1 (NHTSA, 2016b). However, researchers have always predicted that educated and skilled drivers with advanced driver training are less prone to be involved in risky driving; hence, driver training results less traffic crashes (Roenker, Cissell, Ball, Wadley, & Edwards, 2003). Recent studies show that skilled drivers overestimate their capabilities and lean toward indecent driving behavior and habitual over speeding (Allan F. Williams & O'Neill, 1974). Nonetheless, many active safety features, i.e., automatic braking, lane departure warning and parking assistance,

have been installed in vehicles to assist human drivers and reduce human error-related traffic crashes. Since these safety features have been improved, drivers are assured of safe driving conditions and an ensured safe ride. Now researchers and automobile companies are progressing ahead to eliminate human drivers behind the vehicle wheels and bring computerization and automation into the overall transportation system (Antsaklis, Passino, & Wang, 1991).



*Figure 1.* Traffic fatalities per year in the United States

The autonomous vehicle is a global phenomenon, which continues to attract the attention of researchers, the automotive industry, transportation professionals and policymakers worldwide. This vehicle has the potential to become a safe, sustainable,

ecofriendly and personal mode of transportation. An autonomous vehicle can navigate itself on the roads and highways as well as in complex urban traffic scenarios—all without human intervention. The autonomous navigation can avert the crashes currently caused by human error. Fagnant and Kockelman predicted that autonomous vehicles can save more than 21,000 lives per year and eradicate more than four million crashes with a market penetration of 90% (Daniel J. Fagnant & Kara Kockelman, 2015). Furthermore, these advanced vehicles can provide mobility to new road user groups, i.e., children, the elderly and disabled, increase the transportation infrastructure capacity, save fuel and emit fewer pollutants. Autonomous vehicles could drastically change current land use practices by promoting more ride sharing, and reducing the need for parking spots. Vehicle windshields could be used as advertisement billboards! However, researchers predict that the consumers will initially consider these vehicles unsafe and will not spend money to purchase those (D. J. Fagnant & K. Kockelman, 2015). Meanwhile, the Massachusetts Institute of Technology (MIT) AgeLab and the New England Motor Press Association (NEMPA) conducted a survey among nearly 3000 volunteers to explore people's perspectives regarding autonomous vehicles and found that 48% of the participants would never choose the autonomous technology, while 29% do not trust these vehicles to any extent (Abraham et al., 2017). As a result, it is important to conduct a detailed analysis regarding the performance of autonomous vehicles to dispel people's misconceptions concerning these vehicles, and to overcome this big stumbling block on the path of implementing these vehicles.

## **Background and Motivation**

The autonomous vehicle has been ameliorating human lives since 1935—in the pages of science fiction books. The concept of road automation was introduced at the New York World's Fair in 1939 (Geddes & Bel, 1940). The first national automated driving program was started under the Intermodal Surface Transportation Efficiency Act (ISTEA), signed into law in 1991. Later, the Defense Advanced Research Projects Agency's (DARPA's) Grand Challenge was launched in 2005 to motivate the development of algorithms and technologies to develop the first autonomous vehicle that could navigate successfully a route of 132 miles without any driver intervention (Buehler, Iagnemma, & Eds, 2005). The hope was that the autonomous vehicle would be able to replace human drivers in dangerous situations and promised that in the future this vehicle would catalyze a revolutionary advancement in road and highway safety. Since then, many automotive and technology companies have raced to be the first to sell safe autonomous vehicles to consumers. However, these vehicles are equipped with highly tuned sensors and actuators, which are responsible for their autonomous navigation. Despite the many benefits of autonomous vehicles, these advanced components created a new set of challenges. Hence, it is necessary to evaluate these technologies before implementation and to identify strategies to integrate autonomous vehicles into current streams of traffic.

Several states in the U.S. have started to sign new laws and regulations to promote the testing and development of autonomous vehicles. Nevada was the first state to pass legislation on autonomous vehicle testing on state roadways in June 2011. California was the second state with their legislation signed in September 2012 (Nowakowski,

Shladover, Chan, & Tan, 2015). California laws and regulations are applicable for a Level 3 automation system (conditional automation) and higher levels. Automation leveling is based on the definition of the Society of Automotive Engineers (SAE). However, it was mandated that for testing on state roads and highways, each vehicle needs to be equipped with an independent event data recorder (EDR) to record all sensor data that can be gathered at least 30 seconds before a collision happens and to store that data at least for 3 years. Furthermore, original equipment manufacturers (OEMs), who currently hold a permit to test their vehicles on state highways and freeways, must publicly share their test results (reports of incidents, i.e., crashes and disengagement of technology) with the California Department of Motor Vehicles (CA DMV). According to disengagement reports submitted to the CA DMV, various non-autonomous vehicles driven by human drivers were the primary cause for a significant number of incidents (Delphi, 2016; Google, 2016; Mercedes-Benz, 2016; Nissan, 2016; Volkswagen, 2016). Table 1 presents a summary of crashes from recent reports. These reports also include disengagement incidents in which the operator disengages autonomous driving and controls the vehicle manually. About 2,700 disengagements were reported because of unexpected autonomous driving situations such as potholes, poor lane markings, construction zones, and adverse road weather conditions (Fingas, 9 May 2015; Sorokanich, 30 August 2014; Vincent, 13 January 2016). In addition, various hardware and software systems responsible for autonomous driving are prone to disruptions and/or hacking. Researchers recently developed a system consisting of low-power lasers and a pulse generator that can mislead autonomous vehicle sensors, such as LIDAR into seeing objects where none exist (Harris, 4 Sep 2015). Researchers also demonstrated that

hackers could remotely take over the control of autonomous vehicle brakes, accelerators, and other critical safety components (Simonite, 2016). Other researchers recently found an algorithm for autonomous vehicles, which was used to detect objects and was subject to error when traffic signs were camouflaged with stickers, graffiti or art (Evtimov et al., 2017). The researchers examined the algorithm by putting stickers on stop signs and observed that the vehicle misread the sign as a “45 mile per hour” speed limit sign. Moreover, a fatal crash occurred on a state highway in Florida on May 7, 2016 due to vehicular sensors failing to detect a white tractor-trailer while driving in autopilot mode. This report was issued in a preliminary investigations (Klein, 2016). However, the manufacturing company claimed that the vehicular system was designed to assist the driver and that it should not have been left unattended.

Table 1

*California DMV autonomous vehicle crash report*

Automobile Company	Year	Autonomous Vehicle Information	Other Party Information
GM Cruise, LLC	May 2017	Moving	Bicyclist rear ended the autonomous vehicle
Google Auto, LLC	March 2017	Moving	Non-autonomous vehicle rear-ended autonomous vehicle while inching forward with traffic at red light
GM Cruise, LLC	March 2017	Stopped in traffic	Non-autonomous vehicle clipped front of autonomous vehicle while turning
GM Cruise, LLC	March 2017	Moving	Non-autonomous vehicle rear-ended after traffic light turned green

Table 1 (continued)

Automobile Company	Year	Autonomous Vehicle Information	Other Party Information
Google Auto, LLC	Dec. 2016	Moving	Non-autonomous vehicle collided into autonomous vehicle side doors while making left turn
Google Auto, LLC	Oct. 2016	Moving	Non-autonomous vehicle rear-ended autonomous vehicle at a yield sign
Google Auto, LLC	Sept. 2016	Moving	Non-autonomous driver violated red light and collided with right side of autonomous vehicle
Google Auto, LLC	Sept. 2016	Stopped in traffic	Non-autonomous vehicle rear-ended autonomous vehicle while it was yielding to oncoming vehicles
Google Auto, LLC	Aug. 2016	Stopped at stop sign	Non-autonomous vehicle rear-ended autonomous vehicle while it was stopped at stop sign. Driver left the scene of the crash (hit and run).
Nissan North America, Inc.	May 2016	Moving	Non-autonomous vehicle suddenly stopped in front of autonomous vehicle causing autonomous vehicle to rear-end leading vehicle

Potential risks during the transition phase (i.e., from conventional vehicles to 100% autonomous vehicles in the transportation system), as well as the vulnerability of other vehicular and communication technologies, could disrupt the mass deployment of autonomous vehicles on our roads. However, it is essential to conduct a thorough risk analysis of autonomous vehicles. Since the autonomous vehicle is equipped with hundreds of sensors, actuators, and communication devices to navigate autonomously, the reliability of these sensors must be evaluated before mass deployment. Severe traffic crashes may cause and result in fatalities and property losses if the advanced autonomous

vehicle navigation sensors are not fully developed. This risk analysis needs to be designed to explore the root causes of potential failures of autonomous vehicles and identify the chain of events that could lead to the failure of system integrity. The comprehensive risk analysis results are needed to guide policymakers to support the deployment of these advanced vehicles into the large US transportation grid.

### **Research Objectives**

As technology evolves, and continues to evolve each day, the apparent risks associated with these new technologies begin to multiply making the risk analysis process more important than ever. Risk analysis is utilized to identify the potential hazardous sources and accident scenarios and to assess the potential impact these can have on human, environmental, and technological targets. It could become a valuable tool to help risk managers reduce potential threats and policymakers to develop a management and maintenance framework, which a manufacturer must follow to ensure public safety. In this thesis research, a comprehensive risk analysis was conducted to identify the threats associated with autonomous vehicles. Risk analysis is a potential source of novel database information; furthermore, it can guide both professionals and policymakers in their acceptance and regulation of the policies and regulations needed for autonomous vehicle mass deployment. A probabilistic fault-tree analysis tool is used to identify potential risks. Furthermore, it is also necessary to identify strategies to integrate autonomous vehicles into current streams of traffic, as the number of autonomous vehicles will be low at the initial phase. So, this research focuses on the transition phase in which autonomous vehicles will become a part of the current traffic mix of conventional vehicles.



The objectives of autonomous vehicle risk analysis are to:

- 1) Develop the hierarchical sequence of events that may result in the failure of an autonomous vehicle or the infrastructure it depends on, in the form of a fault tree,
- 2) identify shortest routes (i.e., minimal cut sets of the developed fault trees) leading to overall autonomous vehicle system failures and prioritize them based on the failure probabilities of basic event occurrences, and
- 3) simulate a microscopic traffic model and investigate the impact of autonomous vehicle failures on the efficiency of overall transportation infrastructures.

Notably, an autonomous vehicle is represented as equipped with Level 4 and Level 5 automation systems as defined by the SAE (high and full automation). Furthermore, only an autonomous passenger car or a similar vehicle is considered here. Transit trucks or other types of on- or off-the-road vehicles are not included.

### **Organization of Thesis**

This thesis is divided into seven chapters. The first chapter introduces the research, background and motivation behind the research. The objectives of the research are also briefly discussed in this chapter. A thorough literature review on autonomous vehicle systems, risk analysis and its applications, as well as fault tree methods are presented in the second chapter. The detailed methodology is described in the third chapter. The fourth chapter explores the three phases of autonomous vehicle risk analysis to evaluate the reliability of the integration of an automated navigation system. The three phases are: risk identification, risk estimation and risk hierarchization. Then, the results of fault tree analysis are compared and validated with real-world data collected from the California Department of Motor Vehicles (CA DMV). The fifth chapter presents the

Delphi survey structure and the results of the survey. A statistical method called Kendall's coefficient of concordance  $W$  is used to analyze survey results. The algorithm of autonomous vehicle navigation is presented in chapter six. A traffic network is simulated using a microscopic traffic simulation platform and then calibrated. This simulation model is evaluated with different autonomous vehicle market penetration levels. Finally, concluding remarks along with recommendations and future directions are presented in chapter seven. All the codes to run autonomous vehicle simulation on a microscopic traffic simulation platform are provided in the appendices.

## **Chapter 2**

### **Literature Review**

Autonomous vehicles have the potential to eliminate traffic accidents due to human error thereby providing a safe and sustainable transportation system. However, a fully developed autonomous vehicle has not become a reality yet. Even a small misjudgment in autonomous navigation could lead to highly devastating crashes, which could result in both fatalities and property loss. So, a comprehensive risk analysis of autonomous vehicles is required before their mass deployment on roads and highways. Several researchers have developed a preliminary risk analysis model of autonomous vehicles. Different autonomous vehicle sub-systems and/ or a few transportation infrastructure components were considered in those studies; however, weather, other road users (non-autonomous drivers, cyclists, pedestrians, etc.) and road surface conditions were not included. As far as the author knows, the fault-tree based risk analysis of autonomous vehicles in mixed traffic streams considering both vehicular components and infrastructure components has not been conducted yet. This chapter is organized as follows:

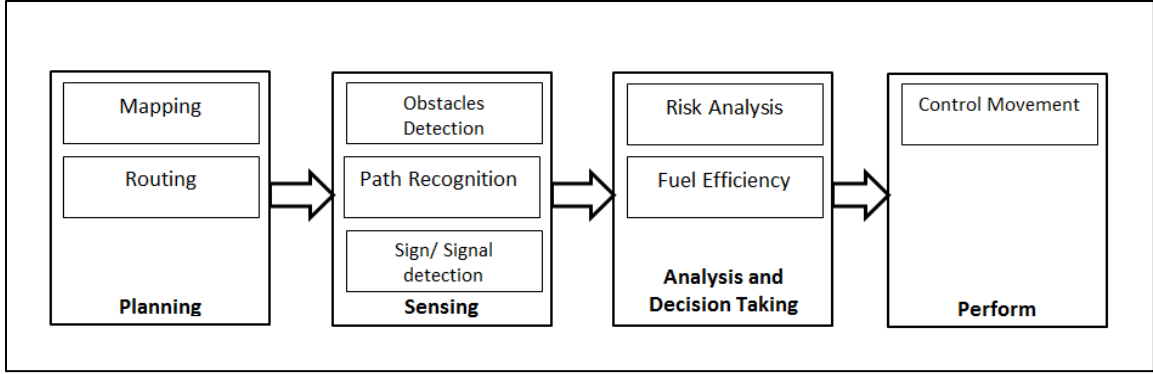
- **Autonomous Vehicles:** This section summarizes the overall system integration of vehicular components used for autonomous navigation. The summary of vehicular components will help identify the potential risks of autonomous vehicles.
- **Risk Analysis:** This section reviews the risk analysis phases, along with analysis elements. Later on, the risk analysis techniques will be classified into two classes. Fault tree method, one of the risk analysis methods, is also explained briefly in this

section, as the author will use this technique to estimate the overall failure rate of autonomous vehicles.

- **Risk Analysis of Autonomous Vehicles:** Finally, the proposed risk analysis methods in different studies will be summarized. These methods will be divided into three separate classes based on their study structures.

### **Autonomous Vehicle**

The concept of an intelligent transportation system has been a subliminal possibility for many futurists since the 1930s. However, the subliminal is finally becoming a reality as Japan, U.S. and Europe focus on large-scale integration and deployment of their individual ITS programs. Connected vehicles, equipped with communication devices and within a connected infrastructure environment, could collect previously unobtainable traffic data and can also share that information with other connected vehicles and monitoring units simultaneously. This communication between vehicles and infrastructure has received a great deal of attention since the 1960s. Transportation officials and engineers have encouraged designers to create higher levels of safety and mobility improvements on the roadways for more than 50 years. The connected infrastructure communication with autonomous vehicles and technologies like global positioning systems (GPSs), drones, and other monitoring devices seem to have unlimited expected benefits. One benefit could come from the implementation of fully autonomous vehicles with the ability to perceive its environment, make route selections, and drive by itself without any human involvement or any occupants at all (Richard Wallace & Silberg, August 2012).



*Figure 2. Infographic architecture of autonomous vehicle functional components*

The exponential growth of processor speeds within the last 30 years and the availability of feasible technologies have enabled transportation engineers to focus on vision-based vehicles detection for driver assistance over last decade. Figure 2 shows different functional components of autonomous vehicles along with their specific tasks. Planning generates potential trajectories for an autonomous vehicle, based on the origin and destination chosen by the passengers. Driver behaviors will be considered while selecting routes on previously loaded maps. Position recognition or sensing detects the surrounding objects, other road users, transportation infrastructure components, and also estimates current positions, attitude, velocity and acceleration. Then, vehicles will optimize and establish safe paths along with other objects to complete a safe trip. Finally, the vehicular control will perform a control movement and the vehicle will be driven to the next position on its trajectories.

Radar-based (S. Park, T. Kim, S. Kang, & Heon, 2003), laser-based (Chieh-Chih, Thorpe, & Suppe, 2003), (John Hancock et al., 1997) and acoustic-based (Chellappa, Gang, & Qinfen, 2004) approaches are being used for vehicle detection, where the system doesn't require powerful computing algorithms and processing units. A camera-based

vision system, which requires fast processing speed and high-capacity data storage, can represent a nearly 360° field of view. This allows a greater handling efficiency when entering a curves, overtaking other vehicles, and early recognition of traffic signals and signs as well as dedicated lanes and bikes or pedestrians (Sun, Bebis, & Miller, 2006).

**Level of automation.** Based on the level of vehicle versus human control, the National Highway Traffic Safety Administration (NHTSA) has specified five levels of automation (Blanco et al., 2015). The goal of these classifications is to provide a common terminology for automated driving. However, SAE International also developed another harmonized classification system for the same purpose. As the level of automation increases, the responsibility of the nonautonomous driver shifts from driving to supervisory tasks. A brief description of the NHTSA autonomous vehicle classifications is given below.

***Level 0 (no automation).*** Over the duration of the journey, the non-autonomous driver solely controls the vehicle (brake, steering, throttle, and motive power), and he or she is responsible for the safe operation of the vehicle. A vehicle may have certain level of driver assistance and support systems (for example: lane departure warning, blind spot warning, etc.). These support systems can provide warnings as well as automated secondary control, like wipers, headlights, hazard lights, etc., but they do not have control over steering, braking, or throttle.

***Level 1 (function specific automation).*** At this level, one or more control specific control functions are integrated into the vehicle system, although the driver controls the overall navigations and motions. The vehicle can cede the authority of the driver and add

certain levels of control in crash-imminent situations, for example dynamic brake support in emergencies.

***Level 2 (combined function automation).*** Automation of at least two primary control functions is involved in this level to release the driver's responsibilities over the control of those functions. However, the driver is still expected to take over responsibility if those assigned controls are perceived to compromise the vehicle's safe operation due to unexpected problems on the roadways.

***Level 3 (limited self-driving automation).*** The driver is relieved from the control of all safety critical functions, although he or she is expected to take over the control occasionally, but within a sufficiently comfortable transition time.

***Level 4 (full self-driving automation).*** The vehicle can navigate, perform all driving control functions, and monitor the roadway for an entire trip without any intervention of the human driver.

On the other hand, there are six levels of automation from "No Automation" to "Full Automation" identified in SAE classification. The comparison between NHTSA and SAE International classifications is shown in Figure 3.

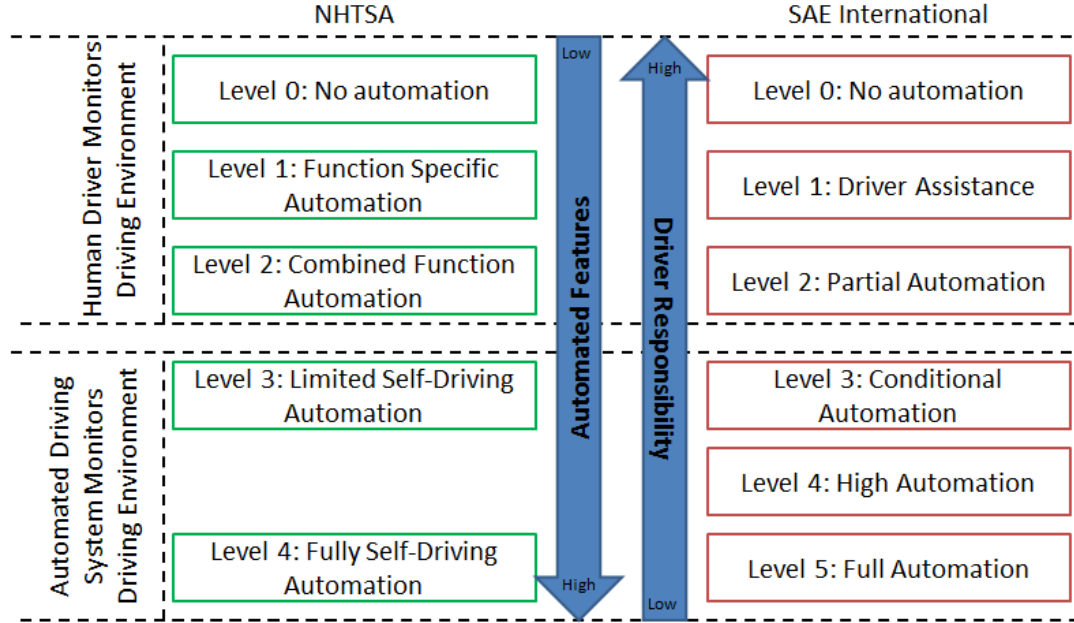


Figure 3. Comparison between NHTSA and SAE International levels of automation classification

It should be noted that this study considers only the Level 4 passenger car for analysis, and this does not include transit or other type of on- or off-the-road vehicles, i.e. trucks, buses, farm vehicles.

**System disintegration.** To identify the potential risks related to a system, the first step is to divide the whole system into basic components. The analysis of technological developments installed in autonomous vehicles could be a way to figure out the sensitive components of these vehicles, which would eventually lead to risk identification. The automotive features which made autonomous vehicles safer than the conventional vehicles are discussed here.

**Intelligent adaptive cruise control system.** Even though road accidents still occur every day with major economic losses to the society, but statistic shows that numbers of fatalities in road accidents are decreasing. There were 1.11 fatalities per 100 million



vehicle miles traveled (VMT) in 2010. This number was 1.53 in 2000 (Congress, 2013). Rear-end collisions were responsible for approximately 1.8 million crashes, which resulted in 1,570 fatalities in 1998 (Persson, Botling, Hesslow, & Johansson, 1999). Moreover, maximum use of highway capacity would be achievable if vehicles could run closely without causing crashes at the posted highway speed (Swaroop & Huandra, 1998; Swaropp & Rajagopal, 1999). But this constant spacing platoon could only stabilize if vehicles are equipped with an adaptive cruise control system (ACC). This system automatically controls the throttle and/or the brake to adjust the vehicle velocity and maintain a predetermined safe distance from the following vehicle. On-board installed sensors, like RADAR and LIDAR, etc., measure the distances between two successive vehicles.

A maximum traffic flow of more than 4200 vehicles/hour per each lane when all vehicles are equipped with this driving assistance system could be achievable (VanderWerf, Shladover, Miller, & Kourjanskaia, January 2002), while manual driving permits around 2000 vehicles/hour (P. Ryus, L. Elefteriadou, R. G. Dowling, & Ostrom, 2011). To evaluate the probability of collision between vehicles, researchers used Monte Carlo simulations and found that ACC significantly reduces collision probability (Touran, Brackstone, & McDonald, May 1999).

Some automaker companies introduced ACC in their cars at the beginning of the 21st century. Researchers found that 1.1 to 10.7% fuel consumption could be reduced by using this driver assistant. Moreover, implementation of safer roadways would be applicable through adopting this system (D. Godbole, R. Sengupta, J. Misener, N. Kourjanskaia, & J.B. Michael, January 1998; W.G. Najm & A.L. Burgett, 1997). A 60%

reduction in air pollution from transportation sectors would be possible if 10 percent of vehicles were equipped with the ACC (Bose & Ioannou, 2001). It requires no road infrastructure modification to work effectively, so this driver assistant system is available for immediate use (T. Chira-Chavala & S.M. Yoo, 1994). Researchers implemented human following behavior based on fuzzy logic or neuro-controllers to train ACC spacing adjustments (Germann & Isermann, 1995). However, nonlinear mathematical control models like sliding mode control (Gerdes & Hedrick, 1997) and optimal dynamic back-stepping control (X. Lu, Shladover, & Hedrick, June 2001) have been used in deriving the desired acceleration for the string stability of the expected vehicle platoon.

***Automotive collision avoidance/ warning system.*** Loss of control causes at least 9 percent of all car crashes in the U.S. every year ("National Motor Vehicle Crash Causation Survey," July 2008). Statistics shows that the drivers' delay in recognizing or judging a "dangerous" situation is responsible for a large number of road accidents. When it is possible to overcome human driver limitations by automating some parts of driving tasks, this type of accident could be eliminated. Researchers have developed one driving assistance system, called the collision avoidance system which requires a RADAR sensor installed at the front of the vehicle. This sensory system could perceive a dangerous situation based on the collection of robust and reliable data, which can be utilized to estimate the time of collision (TTC). If the time to collision at a current speed is lower than the threshold value, then the system automatically controls the car to brake and/or steer from an imminent collision. It was found that more than 50% of rear end collisions could be avoided though collision avoidance system ("Report to Congress on the National Highway Traffic Safety Administration ITS Program," January 1997) and

90% of accidents could be prevented with one second of warning time provided to a driver (Woll, 1997).

The collision avoidance system first warns the driver when the distance between successive vehicles becomes smaller than the warning distance, and in the more critical situation it brakes automatically when this distance drops to less than the braking distance. Honda and Mazda presented a model to determine and scale the warning and braking distance according to drivers' preferences based on different environments (Seiler, Song, & Hedrick, 1998). Later a model was developed to calculate the brake timing for rear end collision warnings (B. Wilson, 2001). Other researchers also eventually developed another nonlinear model, which could derive road tire friction (Kyongsu & Jintai, 2001; Yi, Woo, Kim, & Lee, March 1999). This model was further updated to calculate tire road friction and scale critical distances (Seiler et al., 1998). Researchers also proposed and designed a neural network to estimate the collision avoided path (Eskandarian & Thiriez, September 1998).

***Lane departure warning.*** A considerable portion of road accidents are caused by a temporary and involuntary fading of a driver's vision, which can be caused by sleep deprivation, fatigue, using mobile phone, chatting, or some other diversion, which leads the vehicle to leave its designated lane. In the U.S., about 11% of vehicles that fail to stay in the proper lane cause vehicle crashes ("National Motor Vehicle Crash Causation Survey," July 2008). A machine vision system, called the lane departure warning system, could improve road safety by preventing a vehicle's unintentional deviation from the center of its traveling lane. Different sensors have been researched to perform lane departure warning, including but not limited to LIDAR, camera, and GPS devices. In

case of camera vision-based system, a camera (or multiple cameras) installed on-board visualize the solid and striped markings of the road ahead and then the steering is adjusted to keep the vehicle in the center of the lane.

Researchers used a temporal filter for noise reduction and road marking detecting to diagnose road edges (Beucher & Bilodeau, 1994; Yu, Beucher, & Bilodeau, 1992). Although the detected road edges are typically irregular and rough, this model still requires relatively high computational costs. Another method of lane detection depends on the top view images captured by camera vision, which are compared with the world coordinate of lane edges based on online computation (Bertozzi & Broggi, 1998; Pomerleau, 1995). Deformable mathematical road models are suggested to detach road boundaries based on a linear model which could not provide enough accurate results. Splines or a parabolic model are options, but these models are sensitive to noises (Enkelmann, Struck, & Geisler, 1995; Risack, Mohler, & Enkelmann, 2000). Later researchers developed a model based on particle filtering and multiple cues to be efficient under a variety of conditions like shadows, cloudy days, and rain, but the model could also be applicable to the curved sections of the roadway (Apostoloff & Zelinsky, 2003). An edge distribution function (EDF) was proposed by (J. Lee, 2002) and later modified by (Fardi, Scheunert, Cramer, & Wanielik, 2003) through a boundary pixel extractor to detect curved roads with dashed lane markings. Recently a linear-parabolic lane boundary model was proposed where a linear model was designed to fit the adjacent straight section, and a quadratic function was used to detect incoming curves, even in the presence of shadows and different lighting conditions (Jung & Kelber, 2004).

***Intersection collision avoidance system.*** Recent studies show that 36% of all road accidents in the U.S. occur due to intersection collisions ("National Motor Vehicle Crash Causation Survey," July 2008). In 2004, signal and stop sign intersection crashes are responsible for \$7.9 billion in economic losses (W.G. Najm, J.D. Smith, & M. Yanagisawa, 2007). To avoid this type of collision, an intersection collision avoidance system was designed and developed for predicting driver behaviors at stop sign- and signal-controlled intersections. This new system enables a vehicle to handle emergency intersection problems safely. The vision-based system estimates the time to collision (TTC) in any type of traffic rule violation and controls the speed and acceleration in real time to avoid crashes. However, the DSRC (dedicated short-range communication) system could be used to allocate transmission windows to vehicles approaching an intersection, which starts with generating a poll request to inquire about their maneuver status; then, sends safety messages to ensure safe intersection movements (Rawashdeh & Mahmud, 2008). Inter-vehicular communication leads to a more flexible method for this information communication where all vehicles entering the intersection broadcast their locations with direction, speed and destination (Dogan et al., 2004). Later real time infrastructure communication using telematics and wireless sensor network is proposed to supply base stations with the necessary information for collision prediction and avoidance options (Basma, Tachwali, & Refai, 2011). Magnetic sensors (Kyungbok, Jae Jun, & Dohyun, 2007), the camera vision method (Atev, Masoud, Janardan, & Papanikolopoulos, 2005), radar (Menon, Gorjestani, Shankwitz, & Donath, 2004) and a combination of loop detector and radar systems (Ashkan Sharafsaleh & Chan, November 6-10, 2005) are used as wireless sensors in different research methodologies.

***Electronic stability control.*** Electronic stability control (ESC) systems are another breakthrough driving assistant technology used to monitor the speed of each wheel, the steering wheel angle, yaw rate and lateral acceleration comprising sensors, brakes, the engine control module and a microcomputer. This on-board car safety system is designed to enhance safe driving through improving vehicles' lateral stability and assisting drivers in critical situations or under unfavorable conditions (rain, snow, etc.). When sensor data detect an emergency, the ESC system applies the brakes to individual wheels and possibly reduces the engine torque so as not to lose the control of the vehicles. This system could reduce the number of accidents due to driver error and loss of control. 22 percent of road accidents, which are caused due to running off the edge of the road or a loss of control, could be avoided by ESC ("National Motor Vehicle Crash Causation Survey," July 2008). A Swedish research team showed that ESC could reduce from 20% to 40% of crashes on wet surfaces or surfaces covered by snow or ice (Tingvall, Krafft, Kullgren, & Lie, May 19-22, 2003).

Earlier ESC was treated as an optional driving assistance system on European-U.S. luxury cars. In 1995 the ESC system was first introduced in Europe and later appeared in the U.S. market (Memmer, 2001). Later Audi, Ford, General Motors, Toyota, BMW and Mercedes incorporated this technology into their cars. This system includes sensor offset compensation, sensor signal filtering and processing, sensor plausibility, active wheel lift detection and software enhancement of brake hydraulics to achieve vehicle stability control (Eric Fenaux & Jeremy Buisson, 2007).

A simple model called the  $\beta$ -method was developed to calculate the sideslip angle during traffic maneuvers (Shibahata, Shimada, & Tomari, 1993). By regulating the

engine's torque and wheel brake pressure using traction control components another system was used to minimize the error and help the driver to keep the car under control (van Zanten, Erhardt, & Pfaff, 1995). An on-line sensor monitoring method using sensors in the ESC system was developed, implemented and produced in large volumes (Fennel & Ding, 2000). Later a dynamic model was built and verified using MATLAB and Simulink (Wang & Xue, 2004). A combination of anti-braking and a traction control system was used to derive another dynamic model with control logic for active yaw control (Y. Jia, J. Song, & Sun, 2004). The developed fuzzy logic PID controller is embedded in the modern ESC system to achieve more reliability (Liangmo Wang, Li Tan, Li-hua An, Zhi-lin Wu, & Li, 2012).

***Pedestrian detection system.*** Pedestrian detection is a challenging problem in a vision-based intelligent transportation system using cameras and RADARs installed on fast moving vehicles. Normally, a candidate selection mechanism is used to solve this pedestrian recognition problem in vision based system, which is done by performing object segmentation on either a 3-D scene or 2-D image plane, (Alonso et al., 2007). However, to ensure a low false negative ratio, this system requires yielding lots of candidate per frame and assumes a flat terrain, which causes loss of depth of scene. This system could be successful with less computational cost. A stereo vision system can overcome these problems, but the solution would entail high computational cost and a dynamic calibration model. Infrared images (Fardi, Schuenert, & Wanielik, 2005; Fengliang & Fujimura, 2002) and infrared stereo (Bertozzi, Broggi, Lasagni, & Rose, 2005) have also been applied in different research efforts to provide better visibility at night and during adverse weather conditions.

Researchers mostly use shape analysis to detect pedestrians in real traffic scenarios. Also some other techniques like vertical linear feature with human template (Bertozzi et al., 2003), hierarchical shape templates on Chamfer distance (Gavrila, Giebel, & Munder, 2004), Haar wavelet representation (Mohan, Papageorgiou, & Poggio, 2001), probabilistic human template (Nanda & Davis, 2002), sparse Gabor filters and support vector machines (Hong, Nanning, & Junjie, 2005), graph kernels (Suard, Guigue, Rakotomamonjy, & Benshrir, 2005), and motion analysis (Franke & Heinrich, 2002) have been considered for pedestrian detection in different research papers. The fast and robust algorithm of neural networks has been successfully applied to detect pedestrians and roads in cluttered scenes using a pair of moving cameras (Liang & Thorpe, 1999; Szarvas, Yoshizawa, Yamamoto, & Ogata, 2005).

**Vehicular sensors for automation.** In the previous section, automotive features are used to track down the necessary sensors and components of autonomous vehicles. These sensors have to work smoothly to maintain the autonomous movements of these vehicles; consequently, the failure of one sensor could lead to the failure of the whole system unless there is a backup plan to recover the defective components immediately and automatically. The analysis of the potential risks for each sensor and its reliability as part of the whole system are required to ensure safe transportation systems. To identify these preliminary risks, a fault tree for autonomous vehicles was developed and analyzed to determine system availability or reliability rate (Duran, Robinson, Kornecki, & Zalewski, 2013a). The functional details of these sensors are presented here for further analysis.



**LIDAR.** LIDAR (light detection and ranging) is a remote sensing technology. Its uses are amazingly varied especially since its primary objective is to collect 3D information and to use light in the form of a pulsed laser to measure different distances from its airborne location to earth. LIDAR's role in the autonomous vehicle operation is to collect kinematic information about the vehicle and physical information about its surroundings. The LIDAR optical sensor is installed on the hood of autonomous car. It includes a laser, lens filter, receiver, power regulator, rotating mirror, and onboard processor. The autonomous car LIDAR system is a combination of synchronizing hardware, which includes precision motors and position encoders, as well as an onboard processing unit that detects the objects and produces both 2D and 3D point clouds. The processing unit must be placed at a high clearance location from the ground; moreover, protective measures are needed to protect the unit from foreign object impact, shock or vibration resulting from crashes or rough terrain navigation, which could lead to failure of the system.

High resolution 3D LIDAR could be useful for up to a 50-meter range with efficient operation in the shadows and different lighting conditions (Fishman, 1996). A complex model of roads (Box & Wilson, 1954), precise localization system using GPS and/or an internal measurement unit (IMU) (Au & L.Beck, 2001; Bucher & Bourgund, 1990) are synchronized with the LIDAR system in many research projects for autonomous driving systems. Other researchers combined LIDAR and computer based vision technology for this purpose (Gavrila, 2001).

**RADAR sensor technology.** Radio waves are transmitted into the environment to scatter back information on obstacles around the vehicle and increases awareness of other

vehicles ahead and behind. This sensor keeps a digital eye on other cars and instructs the autonomous car to speed up or slow down depending on the behavior of other drivers. It also assists in the automotive parking feature.

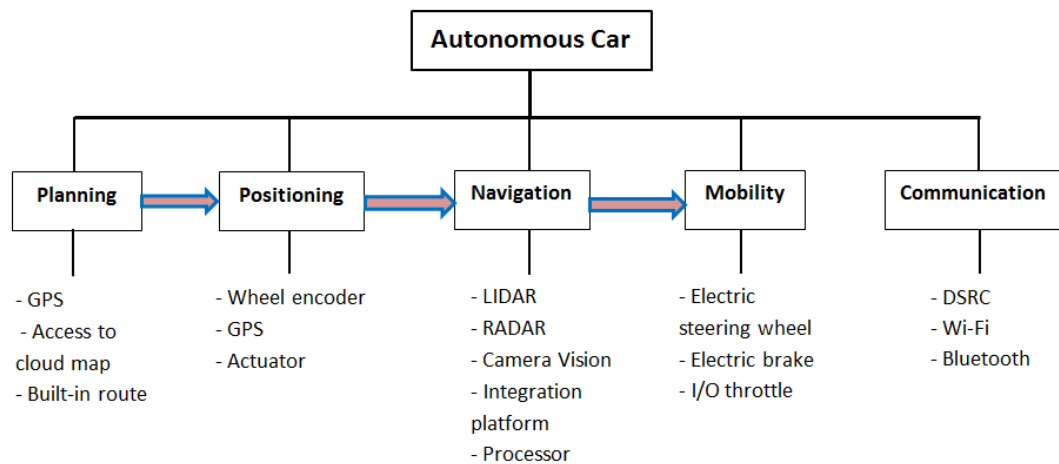
**Camera.** Due to the presence of visual cues and landmarks, the camera-based vision system is used in a variety of research endeavors (Fuke & Krotkov, 1996). Cameras are required in the intelligent transportation system for environment sensing to recognize obstacles with respect to the autonomous vehicle's location and speed. Two-dimensional images using a single camera or 3D maps using dual camera could stereoscopically pinpoint the available space for autonomous movements of the vehicle. These images or maps from the camera vision system are used to extract quantitative information from the scenes, to detect obstacles or to track the targets. The images are segmented into a certain number of pixels. Each pixel is processed and stored, which requires high computational speed and high memory space.

**Global positioning system.** The main sensor used for acquiring navigation and positioning of the autonomous car is the Global Positioning System (GPS) which provides information with one-centimeter precision. To navigate the vehicles autonomously the GPS, with the help of sensors, creates precise maps of the roadway and drives that in the exact direction. This GPS based route tracking could also detect other vehicles on the same roadway and show their exact position on the same scene, as each vehicle has a GPS receiver (Goel, Dedeoglu, Roumeliotis, & Sukhatme, 2000). Due to signal disturbance and other interference from the atmosphere the position estimated using a GPS may be off by several meters. Also, tall buildings obstruct the satellite

signal. Fuzzy variables and rules are used to model the guidance system and correct the computed trajectory (Y. C. Lee, June 1986).

**Wheel encoder.** The wheel encoder is used to keep track of an autonomous car's direction, speed and the distance a wheel travels. It could be helpful for precise movement as it could allow the vehicles to turn exact angles or move exact distances. It proves its high efficiency in planar environment as a dead-reckoning sensor, but is not applicable when there is significant deviation from planar motion (Lapp & Powers, 1977). This sensor could assist for reverse parking through navigating into a tight parking spot when the car is engaged in reverse gear.

A brief info-graphics showing the significance of different autonomous vehicle components and their functions are presented in Figure 4. Also, a summary review on autonomous vehicle functions and sensors is presented in Table 2.



*Figure 4.* Functional responsibilities of vehicular sensors

Table 2

*Development of autonomous driving assistance technology*

Accident causes	% of crashes in the US	Potential Solution	Sensors	Applied Algorithms	Benefits/Improvements
Rear end collision due to uncontrolled driving, monotony driving, fatigue driving on long trips	Approx. 1.8 million crashes in 1998 (S. Park et al., 2003)	Intelligent adaptive cruise control system	RADAR LIDAR	Fuzzy logic or neuro-controllers (Chieh-Chih et al., 2003)	<ul style="list-style-type: none"> <li>- Reduced rear-end collisions</li> <li>- Reduced fuel consumption (1.1 to 10.7% achievable)</li> <li>- Maximized use of highway capacity (John Hancock et al., 1997)</li> </ul>
Drivers' delay in recognizing or judging the "dangerous" situation	Loss of control leads to at least 9 percent of all car crashes (Chellappa et al., 2004)	Automotive collision warning/avoidance system	Camera	Neural network (Zehang, Bebis, & Miller, 2002)	<ul style="list-style-type: none"> <li>- Reduced crashes</li> <li>- Critical situations handled safely and precisely</li> <li>- Automatic braking</li> </ul>
Temporary and involuntary loss of a driver's vision by falling asleep, fatigue, using mobile phone, chatting, etc., which leads the vehicles to leave their designated lane	About 11% of vehicles failed to stay in the proper lane to cause vehicle crashes (Chellappa et al., 2004)	Lane departure warning	Camera Global positioning system	Particle filtering (Ponsa, Lopez, Lumberras, Serrat, & Graf, 2005) Edge distribution function (Onieva, Alonso, Perez, Milanes, & de Pedro, 2009)	<ul style="list-style-type: none"> <li>-Reduced crashes</li> <li>-Prevention of unintentional deviation of vehicles from the center of road</li> <li>- Detect road edges even in extreme lighting conditions (Wo, x, hler, &amp; Anlauf, 1999)</li> </ul>

Table 2 (continued)

Accident causes	% of crashes in the US	Potential Solution	Sensors	Applied Algorithms	Benefits/Improvements
Drivers' misjudging the traffic signs and signals, or disobeying them after approaching intersection	36% of all road accidents (Chellappa et al., 2004)	Intersection collision avoidance system	Camera vision Loop detector RADAR	Neural network	-Reduced intersection collisions - Safe intersection movements
Lack of speed control while driving, inappropriate steering wheel angle, unsafe driving under unfavorable conditions	Almost 22 percent of road crashes (Chellappa et al., 2004)	Electronic stability control	Wheel encoder LIDAR RADAR	Fuzzy logic PID controller (Lagadec, 1980)	- Reduced crashes - Improved lateral stability of vehicles in extreme conditions
Unsafe pedestrian road crossings, inattentive driving, delay in response	14.80 % of total accidents (Porter, 1981)	Pedestrian detection system	Camera vision Infrared sensors	Shape analysis, Probabilistic human template (Porter, 1981), Gabor filters and support vector mechanics (Porter, 1981), Neural networks (Hany H. Ammar, November 2000)	-Detected pedestrian movements - Guided the vehicles to a safe route based on pedestrian movements

## **Risk Analysis**

The development of risk management processes has become a topic of treat concern recently as industries and businesses worldwide attempt to overcome potential threats and ensure the safety of their systems. This risk analysis is used in many different fields, including but not limited to: industrial plant design (C. Alonso, 1998), construction project managements (Chapman, 1997; Ross & Donald, 1995), toxic goods transport (Gadd, Leeming, & Riley, 1998; Tiemessen & Zweeden, 1998), hazardous site management performance (Hurst, 1996), medical records and management (Bogen, 1990) as well as software management (Boehm, 1991). In these risk analysis approaches, the dynamic behavior of a system is considered because the active components can be sources of failure and unexecuted fault prevention will result in failure (S. Yacoub, November 1999).

**Phases of risk analysis.** Risk analysis can be performed at various development phases and can guide future research for better safety in this field. There are three main phases researchers consider in risk analysis methodologies (D. White, 1995). They are:

- Risk identification
- Risk estimation
- Risk hierarchization

Risk identification consists of two interconnected tasks, 1) disassembling the whole system into small parts to make the process easier to understand and 2) examining the behavior of those small parts. Components can range from a simple sensor, an actuator, or the integration platform and database system to links between infrastructures and vehicles. Once the components are identified, the failure rate of each vehicle part

used for automation is determined to describe and quantify the risks related to the whole system. Later in the second phase, these probabilistic failure values will help autonomous vehicle design and maintenance engineers to estimate the significance of these risks as a whole system and determine the system reliability. The ranking of these failure events obtained through all the work completed up to this point is the aim of the last phase, which is hierarchization. This phase detects the shortest possible route(s) to lead the top event failure.

**Elements of risk analysis.** The three phases relate to each other by three elements which are essential to carrying out the risk analysis. They are available input data, expected output data, and selected method. After risk identification, the potential failure probabilities concerning the studied autonomous vehicles system are collected. There are seven classes of input data used by different researchers, which are:

- [1] Plans or diagrams: The details of industry floor plans, i.e., production sites and storage units are used in this class.
- [2] Process and reactions: Mechanical and chemical features of the system, operation requirements and kinetic parameters are considered as process and reaction inputs.
- [3] Substances: Physical and chemical properties of materials, material quantities and their toxicological information are used as substance inputs.
- [4] Probability and frequency: System reliability, failure types, frequency of failure and time dependent failure rates are used as probability and frequency input data.

[5] Policy and management: Safety rules and regulations are included as an input model in this type of fault tree; examples could include but are not limited to: transport safety requirements, operational safety and safety management.

[6] Environment: Topological data and surrounding information are used as input data in this class.

[7] Text and historical knowledge: Historical information and previous analysis results are included in this input class.

Including recommendations as an output of risk analysis, the outputs could be classified into four categories. They are:

- 1) Management: The outputs of this category are recommendations, modifications and updated operational procedures.
- 2) Lists: The lists of hazards, domino effects, errors, failure and damages, failure causes, critical activities and accident scenarios are the outputs.
- 3) Probabilistic: The system failure rates, system reliability performances and accident frequencies are generated as the results.
- 4) Hierarchization: The severity, system criticality, performance index and organization index are considered as the outcomes of risk analysis.

The next step is the selection of a method, where there are two types of methods. These types could be divided into three categories based on the approach selection (J. Tixier, G. Dusserre, O. Salvi, & Gaston, July 2000). They are:

- Qualitative: deterministic approach, probabilistic approach and combination of deterministic and probabilistic approach



- Quantitative: deterministic approach, probabilistic approach and combination of deterministic and probabilistic approach

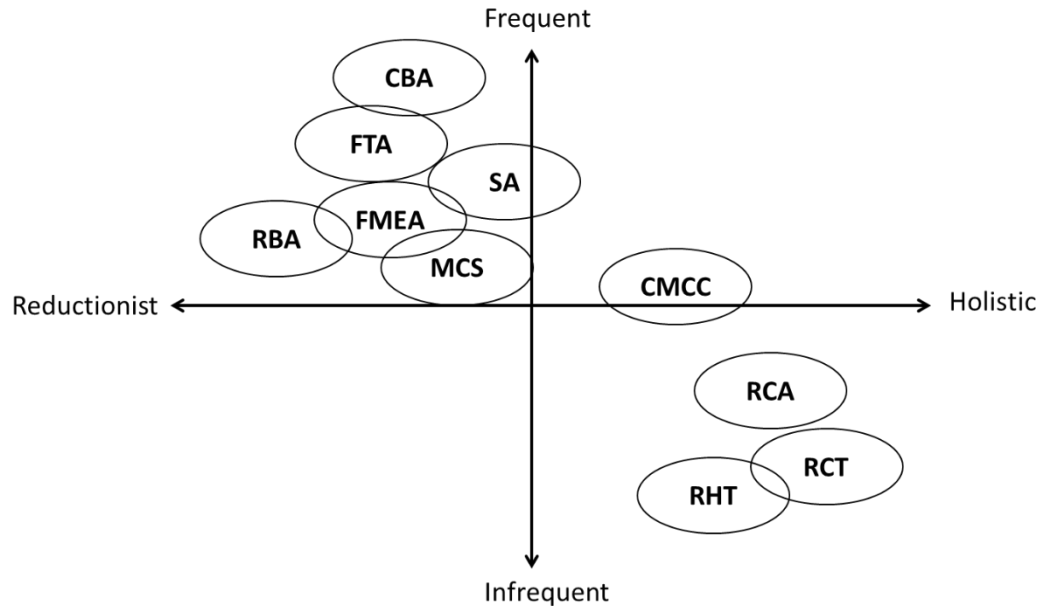
The consequences and their products, the equipment and quantification of impacts on human, equipment and environment are considered in deterministic approach. The deterministic approach can be performed qualitatively and quantitatively. On the other hand, the frequency of hazardous situations and potential occurrence of those hazards is considered in probabilistic approach. Similarly, this probabilistic approach can be conducted qualitatively and quantitatively.

**Classification of risk analysis techniques.** There are numerous techniques and methods used in risk analysis. However, risk analysis techniques can be classified into two types (J. Tixier et al., July 2000). They are:

1) Holistic Techniques: In this category, risk analysis techniques consider the multiple partial views of the problem's environment. A systematic upward movement is carried out here to analyze the overall risk probability of the system. Thus, the risk probabilities study can include: risk compensation theory, root cause analysis, risk homeostasis theory, etc.

2) Reductionist Techniques: This category breaks down the overall system into simplest parts and estimates the impact of those parts on the overall risk analysis of the system. Fault tree analysis, cost benefit analysis, ontology-based analysis, Monte Carlo simulation, failure mode and effects analysis, etc. are examples of reductionist techniques.

The risk analysis methods are plotted in Figure 5, a two-axis figure, based on their classifications and frequency of use. This figure suggests that reductionist methods are more frequently used in nature.



*Figure 5.* Classification of risk analysis methods: cost benefit analysis (CBA), fault tree analysis (FTA), stability analysis (SA), risk benefit analysis (RBA), Monte Carlo simulation (MCS), failure mode and effects analysis (FMEA), common mode common cause (CMCC), root cause analysis (RCA), risk compensation theory (RCT), and risk homeostasis theory (RHT)

### **Risk Analysis of Autonomous Vehicles**

Risk analysis of autonomous vehicles identifies undesirable events and sequences of events leading to autonomous navigation failure, which could lead to road crashes, passenger fatalities, pedestrian injuries, vehicle damage, and external property damage. Researchers followed different paths to assess the potential risks related to autonomous vehicles. Risk analysis methods utilized for estimating the success rates of autonomous

navigation can be categorized into three different classes. They are: i) situation based analysis, ii) ontology based analysis and iii) fault tree based analysis.

**Situation-based risk analysis method.** The process of analyzing newly identified risks or threats based on the solution of similar previous problems is called the situation-based risk analysis method. It is assumed that a complex driving situation can be reported by entities, their attributes, and their connections among each other. In this method, driving situations are described as traffic-oriented factors collected over temporal and spacious patterns. A baseline model is developed to store the prior knowledge of relevant situation-specific concepts as templates. The checklists of risk and their factors are stored based on integration of background knowledge, and they describe complex risk situations in a comprehensive way. Then, the risk identification is carried out as an ongoing risk management task to accomplish the success of an endeavor. Situation-based risk assessment method can be grouped into five steps. They are:

- (i) Specification of risks related to autonomous navigation: Risk situations are defined by using the entities and their inter-relationships based on expert background knowledge of previously explored incidents. The relational dependencies need to be evaluated. The collision between the autonomous vehicle and other road users could be an example of identified risk situations in road surrounding environment.
- (ii) Definition of model concepts: After identification of risk situations, the attributes and their inter-relationships must be defined using object-oriented probabilistic relational language. For example, the risk probability of collision with other road users can be defined as a function of their distances and relative velocities.

- (iii) Construction of world model: The identified risk situations are then transformed into a world model to represent the actual state of the world in terms of entities and relationships.
- (iv) Construction of probabilistic network: A graphical network of probabilistic attributes and their casual dependencies is generated to reason about the current driving situations.
- (v) Assessment of current situation: In this final step, the expected inferences are defined and addressed based on the developed probability distribution.

The identification of risks related to autonomous vehicles and the reasoning behind driving situations have been prioritized by researchers in previous years. To estimate the domain of driving situations, Monte Carlo simulations were used in the case of rear-end collisions (Hillenbrand & Kroschel, 2006). Hidden Markov models (HMMs) were utilized to model complex situations in (Meyer-Delius, Plagemann, & Burgard, 2009), although a new HMM had to be assigned for each situation. Laugier et al. updated the risk analysis for simple traffic scenarios by combining the Hidden Markov Model and Gaussian Process Model (D. White, 1995). The later use of Markov logic networks were improved to describe domains as interconnected objects for driver assistance systems and specified the model as more compact and thus modular (Stiller, Kammel, & Lulcheva, 2008). Researchers also deployed a knowledge-based risk analysis framework to develop simple risk patterns for autonomous vehicles using data collected by vehicle sensors. Then, risk values were evaluated (Bogen, 1990; Swaropp & Rajagopal, 1999; VanderWerf et al., January 2002). Other external sensors like RADAR (Jocoy & Knight, 1998), GPS position sensors (Miller & Qingfeng, 2002), wireless communication (Jihua & Han-Shue, 2006) and cameras (Amditis et al., 2010) were used to predict collisions

and warn the driver in case of threats. Collision risks were also predicted based on intersections of future trajectories, and different shapes of overlapping regions considered in different studies. These risks were identified by sets of points (Batz, Watson, & Beyerer, 2009), circles (Ammoun & Nashashibi, 2009), polygons (Broadhurst, Baker, & Kanade, 2005), etc. Then risk probabilities were estimated using the percentage of overlap between the trajectories. Also to generate better predictions of vehicle trajectories on curved roads, the differentiable continuous curves were adopted in (Katrakazas, Quddus, Chen, & Deka, 2015).

Physical parameters of vehicles were considered for developing a risk assessment platform for safe motion planning (D. White, 1995). In (J. D. Lee, M. L. Ries, D. V. McGehee, & Brown, 2000), traffic situations affecting one road user were broke down into sets of attributes, which were linked using a Bayesian network. However, these sets were separated from each other, and the separation created issues while propagating the effect from one set to another. Some other researchers allowed the interactions between the sets of attributes to resolve this problem. Although Vacek et al. developed a model using situation-based reasoning, their model could fail due to an excessive number of situations in the model base (Vacek, Gindele, Zollner, & Dillmann, 2007). Different algorithms were utilized to predict obstacles on the vehicle trajectories for both intersection and non-intersection segments, including but not limited to: game theory (Martin, 2013), mixed-observability MDP (Meyer-Delius et al., 2009), and multiple criteria decision making (Furda & Vlacic, 2011). However, this risk analysis method is computationally expensive and the success in risk estimation depends on the correct

prediction of vehicles' future trajectories. One example of a situation-based autonomous vehicle risk analysis is shown in Figure 6.

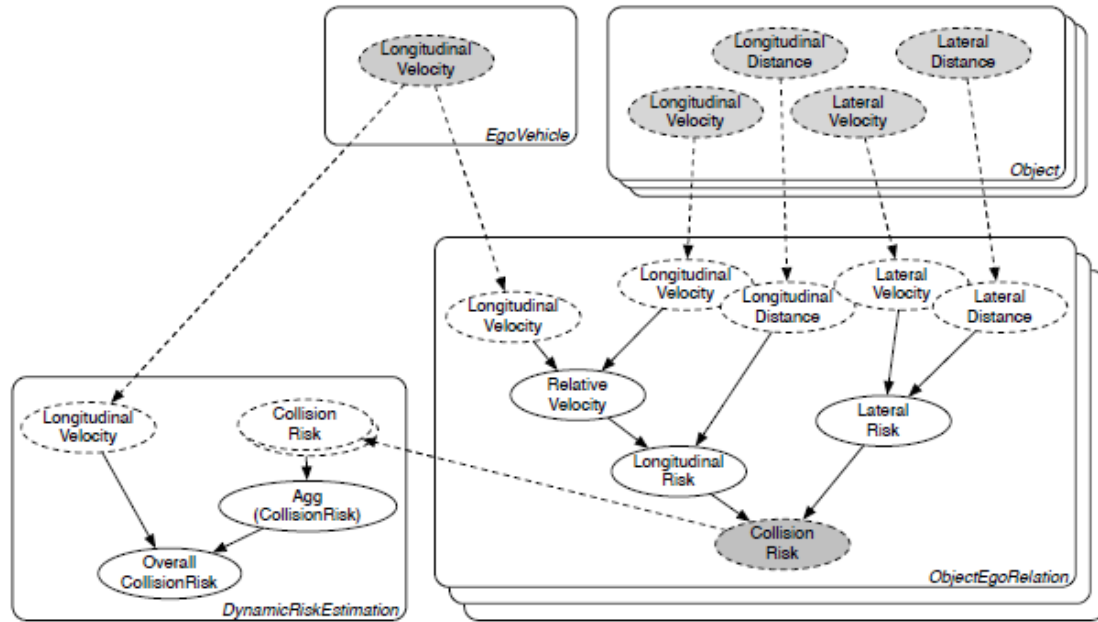


Figure 6. An example of situation-based collision risks identification and evaluation (Hurst, 1996)

**Ontology-based risk analysis method.** Ontology is defined as the specification of a conceptualization of domain knowledge. It is the hierarchical semantic network of basic entities and their inter-relationships based on a corpus of texts. In ontology, a terminological box (TBox) carries the concepts of the domain. The TBox contains basic attributes, their relationships and rules as well as constraints on attributes. Instances of attributes and roles among such instances stay within an assertional box (ABox). Real world data and attribute properties can be stored in this box. A language used to represent the background knowledge in ontology is called description logic (DL), which is a subset

of the first order predicate logic. There are several tools available, such as PROTEGE and SWOOPS, which can edit and verify ontology consistencies.

Researchers applied the ontology-based reasoning for risk analysis of an autonomous vehicle, as it is well suited for modeling multi-parameter traffic situations and also for performing logic reasoning. Complex traffic situations, like intersection traffic signal cycle times and phases, were modeled in the ontology (Keyarsalan & Montazer, 2010; Pommerening, Wöfl, & Westphal, 2009). This method was proposed and successfully utilized to represent different behaviors and depict the interactions between the attributes of road surroundings without stability issues (Armand, Filliat, & Ibañez-Guzman, 2014; Hülsen, Zöllner, & Weiss, 2011; Pollard, Morignot, & Nashashibi, 2013). The driver's ability, road surroundings and vehicle performances were considered for modeling automated ground vehicles risk analysis (Pollard et al., 2013). Another ontology model was proposed to deduce the risks for autonomous navigation due to pedestrian behaviors (Armand et al., 2014). However, it was assumed that pedestrians and control vehicle will obey the traffic rules, which is not valid in the real world. Several sensors were maintained to acquire the information related to the road attributes, like a camera, radar, GPS, ultrasonic sensors, etc. It is preferred to enrich the data by using multiple sensors simultaneously, but the high price of multiple sensors, installation complexity and computation load could be a drawback. The footage captured from the driver's perspective using a monocular camera were utilized in the proposed ontology-based framework by Mohammad et al. (Worrall, Orchansky, Masson, & Nebot, 2010). The proposed ontology framework in this study is shown in Figure 7. The pedestrian

behaviors in different traffic scenarios were examined here; however, the authors did not consider other road users, weather conditions, and road surfaces.

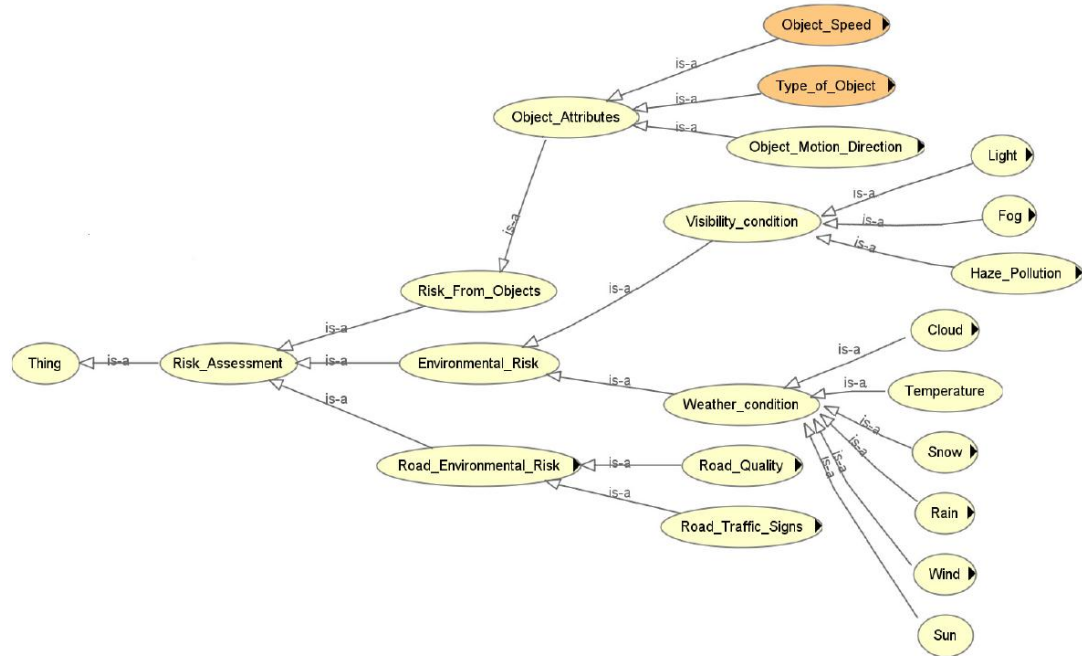


Figure 7. An example of ontology structure for autonomous vehicle risk analysis (Worrall et al., 2010)

**Fault tree-based risk analysis method.** The fault tree determines the potential causes of an undesired event, which represents a safety hazard or economic loss. It is suitable for a nonrepairable system where the failure of components is independent (Ma & Trivedi, 1999). This risk analysis method was proposed by the former AT&T Bell Laboratories (now Nokia Bell Labs), and was initially applied in the aerospace industry (F. I. Khan & Abbasi, 1998). The fault tree analysis method encourages analysis of how a particular component can impact the overall performance of a system and identify the causes of undesired events (Ansell & Wharton, 1992; Ballard, 1992; Wilson & H. C &



Keller, 1990). However, to understand the cause-effect process, a thorough review of the overall system is required to conduct an effective analysis (Vesely, 1984). Therefore, the fault tree starts from a complete system failure and moves backwards to identify all possible causes. A graphical technique is used to represent the fault tree structure where all components are branched off based on their interconnections with top level system failure. These branches are assumed to be independent of each other, i.e., mutually exclusive events (Bell & E, 1989). The analysis method has the ability to identify the shortest route (i.e., minimal cut-sets) to failure of the top-level event. However, some limitations of this risk analysis system must be recognized (Yllera, 1988), as the reliability and failure data of components of the fault tree are required in the analysis and these data control the accuracy of the analysis. To overcome this problem the researchers proposed fuzzy mathematics to reduce the dependency on component failure data (Rauzy, 1993).

Nowadays, this model is commonly used to evaluate the reliability of complex systems in many fields, both qualitatively and quantitatively, such as the systems found in nuclear reactors and petrochemical industries (M. A. Chowdhury, Garber, & Li, December 2000; Greenberg & Cramer, 1991; Lees, 1996; Qingyou & Hao, 1999). After the Challenger incident in 1986, the National Aeronautics and Space Administration (NASA) emphasized performing quantitative risk or reliability analyses using the fault tree method for its space missions' safety assessments. The US Nuclear Regulatory Commission developed a handbook on fault tree construction and evaluation in 1981, and this manual has been considered as the leading technical document on fault tree application (Vesely, Goldberg, Roberts, & Haasl, 1981). Besides, this method has been

used in various other fields, such as aircraft design processes (Volkanovski, Cepin, & Mavko, 2009), vehicular navigation failures (Bhavsar, Das, Paugh, Dey, & Chowdhury, 2017), nuclear power plant design (C. Alonso, 1998), industrial plant designs (Davis-McDaniel, Chowdhury, Pang, & Dey, 2013; W. P. G. Schlechter, 1996), bridge failure analysis (Chapman, 1997), construction project management (Tiemessen & Zweeden, 1998), toxic goods transport (Hurst, 1996), hazardous site management (Bogen, 1990), and medicine (Ammar, Cukic, Mili, & Fuhrman, 2000).

Recently, researchers have utilized fault trees to analyze the impact of autonomous vehicle sensor failure on overall system success rates. In addition, the autonomous vehicle features solely responsible for turning a traditional vehicle into an autonomous vehicle has been evaluated using the fault tree analysis. Swarup and Rao disassembled the adaptive cruise control (ACC) system of an autonomous vehicle and investigated the causes of failures using the fault tree analysis method (Swarup & Rao, 2014). RADAR and the speed sensor, two very important components of ACC system, were explored in this study and broke down into basic potential hazards. However, the authors only considered qualitative risk assessments of the ACC system and did not estimate the failure probability value of the overall system. Duran and Zalewski investigated the causes and effects of failures related to LIDAR and the camera-based computer vision system (Duran, Robinson, Kornecki, & Zalewski, 2013b). To estimate the failure probabilities, the Bayesian belief network was modeled, and the Netica tool was used for this purpose (Norsys). Figure 8 shows the fault tree's graphical representation as developed in the referenced study.

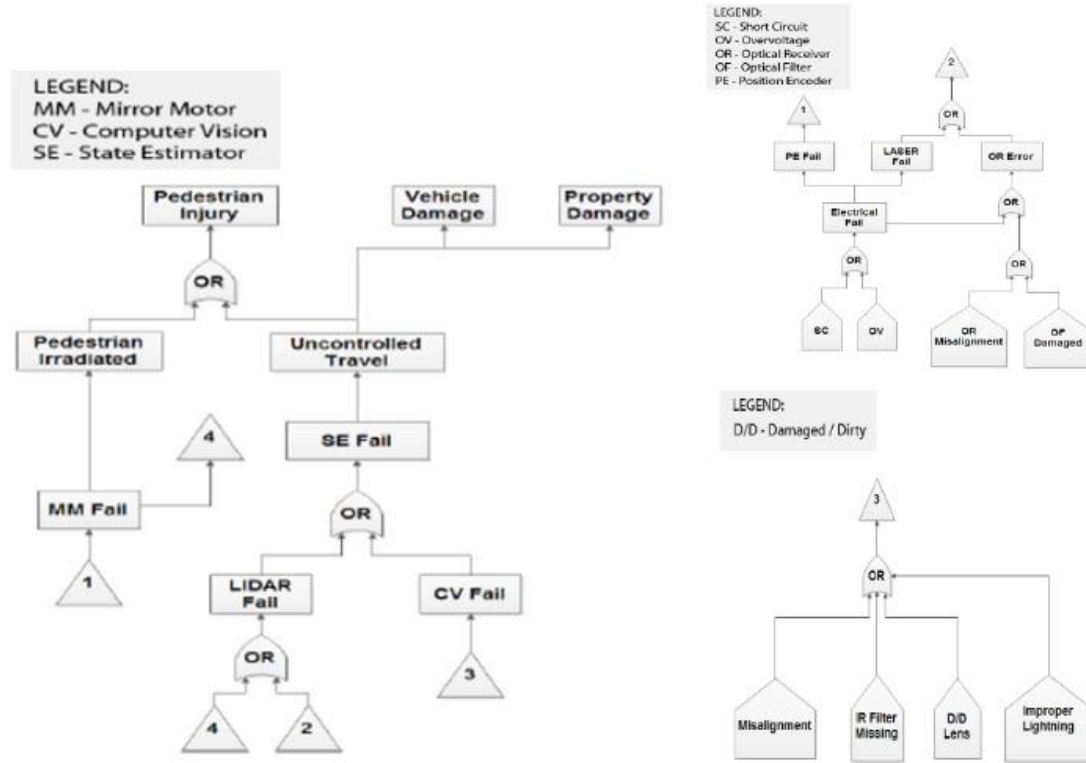


Figure 8. A sample graphical representation of fault tree analysis (Duran et al., 2013b)

Researchers have identified different road variables, which could impact autonomous navigation, but the combined impact of all the different vehicular equipment, other road users, and infrastructure components was not investigated. The overall summary of different approaches conducted so far is sketched in Table 3. In this research, the fault tree analysis method was used to investigate the combined impact of vehicular components and transportation infrastructure component failures. The mathematical derivations of the fault tree method are described in the next section.

Table 3

*Summary of risk analysis techniques used*

Analysis Types	Authors	Parameters Considered	Algorithms	Limitations
Situation Based	Hillenbrand et al., 2006 (Hillenbrand & Kroschel, 2006)	Rear-end collision and crossing collision at intersection	Monte Carlo	- Only applicable in case of simple intersections - Risks from vehicular components were not considered
	Laugier et al., 2011 (Laugier et al., 2011)	Collision risk assessment based on multiple sensors data	Hidden Markov Model and Gaussian Process	- High prices of multiple on-board sensors - High computation power required for parallel processing
	Martin, 2013 (Martin, 2013)	Interaction with other drivers on multilane highways	Game theory	- Only valid when each drivers knows all possible trajectories and destinations of other drivers
	Platho et al., 2012 (Platho, Groß, & Eggert, 2013)	Road users and surrounding entities affecting users	Bayesian network	- Entities were separated from each other - Could fail in complex situations with multiple entities
	Furda and Vlacic, 2011 (Furda & Vlacic, 2011)	Attributes based on priori information, sensor measurements and V2X communication	Multi-criteria decision making (MCDM)	- Limited driving maneuvers were considered here - High computational power required for real-time decision making
Ontology Based	Armand et al., 2014 (Armand et al., 2014)	Different relationships between design vehicle and various road entities (pedestrians, other vehicles, infrastructures, etc.)	Ontology framework	- Limited real time applications - Depends on the frequency of GPS receiver - Not compatible for every driving scenario, only applicable when meeting entities already defined in system

Table 3 (continued)

Analysis Types	Authors	Parameters Considered	Algorithms	Limitations
Ontology Based	Hulsen et al., 2011 (Hülßen et al., 2011)	Roads, lanes, traffic signs, traffic lights, and other road users	Ontology framework	<ul style="list-style-type: none"> <li>- Fixed road geometry was considered without incorporating uncertainties</li> <li>- Qualitative analysis</li> <li>- Did not evaluate in real-world; only tested in simulation</li> </ul>
	Pollard et al., 2013 (Pollard et al., 2013)	Vehicle perception, visibility, weather, traffic signs and road types.	Ontology framework	<ul style="list-style-type: none"> <li>- Separate model based on level of automation</li> <li>- High computational power required</li> </ul>
	Kaloskampis et al., 2015 (Mohammad, Kaloskampis, Hicks, & Setchi, 2015)	Estimation of risks related to pedestrian behaviors using Camera feeds	Ontology framework, Gaussian mixture model	<ul style="list-style-type: none"> <li>- Other road users, weather conditions and road surfaces were not considered in study</li> <li>- Data from video feeds will require high computational power</li> </ul>
Fault Tree Based	Swarup and Rao, 2015 (Swarup & Rao, 2014)	Identification of potential threats of Adaptive Cruise Control	Fault tree	<ul style="list-style-type: none"> <li>- Qualitative analysis</li> <li>- Impacts of each cause were not ranked</li> </ul>
	Duran and Zalewski, 2013 (Duran et al., 2013b)	Risks associated to LIDAR and Camera vision was investigated	Fault tree and Bayesian belief networks	<ul style="list-style-type: none"> <li>- Other vehicular components were not included</li> <li>- Limited to vehicular components</li> </ul>

## Fault Tree Analysis Structure

The fault tree is developed by disintegrating the overall system into its subsystem failures, which later breaks down into lower resolution events. This process continues until no further disintegration can take place. These terminating events are called “basic events.” The failure of the overall system is referred to as a “top-level event,” and the other events are linked to the top-level event with its basic events at the bottom, which are called “intermediate/ casual events.” The top-level event and basic events are interconnected based on the hierarchical and logical relationships between events that lead to the failure of the top event. The schematic of the fault tree in Figure 9 shows these logical relationships presented as “Gate.” The “AND” and “OR” gates are widely used to illustrate the relationship between input and output events.

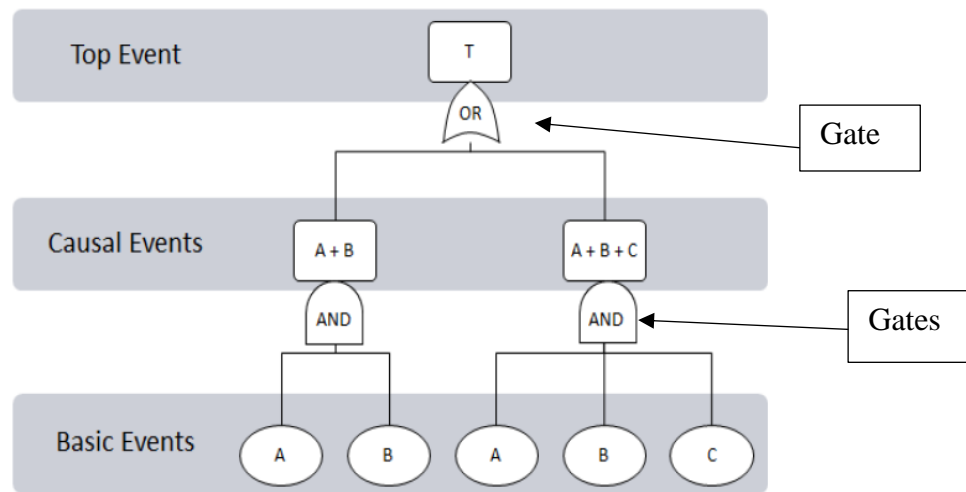


Figure 9. An example of fault tree structure

Besides the basic and intermediate events, undeveloped, conditional, and house events are also used while developing the fault tree in many research projects. Different

events along with their symbols and description are shown in Figure 10, which also uses EXCLUSIVE OR, PRIORITY AND, INHIBIT, and TRANSFER gates in specific cases.











Gate Name		Description
	OR	Output event occurs if one of the input events occurs
	AND	Output event occurs if all the input events occurs
	EXCLUSIVE OR	Output event occurs if one but not both of the input events occurs
	INHIBIT GATE	Output event occurs if a single input event and a conditional event occur
	PRIORITY AND	Output event occurs if all input events occur in a specific sequential order
Event Name		Description
	INTERMEDIATE	An event that results from one or more preceding events acting through logic gates
	BASIC	Initiating event which cannot be further developed
	UNDEVELOPED	An event that cannot be further developed due to insufficient information
	CONDITIONAL	A specific condition applied to an INHIBIT or PRIORITY AND gate
	HOUSE	An event which is expected to occur or not occur

Figure 10. Fault tree gates and events

**Fault tree mathematical formulation.** In this study the logical relationships are restricted to “OR” and “AND” gates. An OR gate represents events that are mutually exclusive events, where one of the preceding events could lead to the failure of the overall system. In “Set Theoretic” terms, this is equivalent to the union of the basic and intermediate events. The probability of the OR gate output can be formulated as follow:

$$P(X \text{ OR } Y) = P(X \cup Y) = P(X) + P(Y) - P(X \cap Y)$$

On the other hand, an AND gate represents a combined failure of all events required to lead to a whole system failure. This gate is related to the intersection of two sets in the “Set Theory.” The mathematical formulation of AND gate is given below:

$$P(X \text{ AND } Y) = P(X \cap Y) = P(X) \times P(Y)$$

### **Summary**

In summary, researchers conducted the risk analysis of an autonomous vehicle using situation-based and ontology-based methodology. However, these risk analysis studies were focused on vehicular surrounding components. Not a single study was done on vehicular components until then. Recently, a fault-tree based risk analysis was completed considering the failure of autonomous vehicles due to vehicular components failure. However, this study failed to include the failures of infrastructure components in the autonomous vehicles risk analysis.



## **Chapter 3**

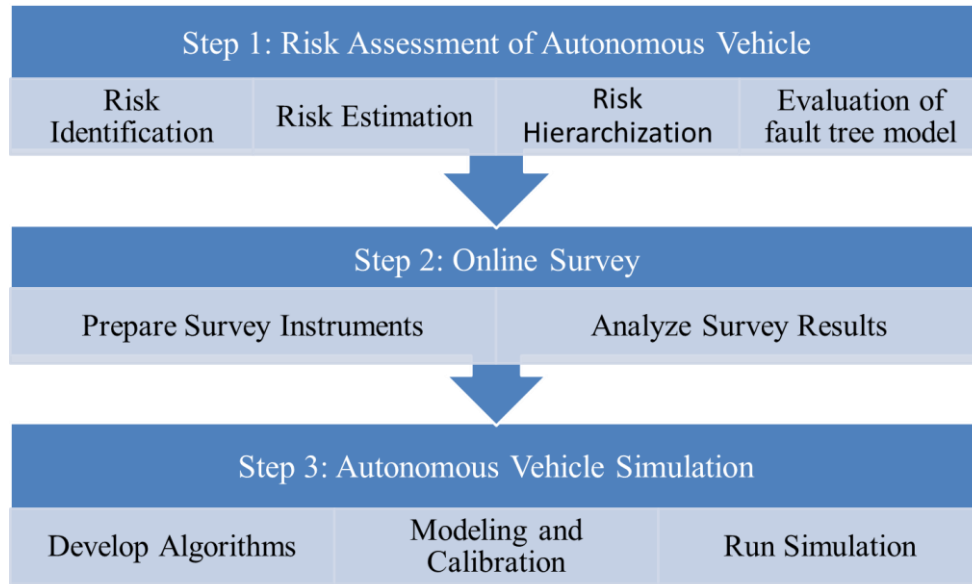
### **Method**

The research method featured in this paper focuses on utilizing the fault-tree analysis approach to identify risks associated with autonomous vehicle failure when it is sharing the roadway with other conventional vehicles driven by non-autonomous drivers. Later, an online survey was conducted to justify the failure probabilities of components through the literature review. Furthermore, a traffic micro-simulation platform was utilized to determine the safety and operational impact of autonomous vehicle deployment in a mixed traffic stream. This chapter presents the detailed approach adopted for this research. A summary of the overall research method is explained in the first section. Three separate but interconnected steps are followed in this research. These steps are explained in three consecutive sections. The first step, risk assessment, is described in Section 3.2. This step is further grouped into three sub-tasks. In the next section, the online survey structure is explained. Finally, Section 3.4 concentrates on traffic simulation development.

### **Research Method**

The overall research method was divided into three major steps. The flow of the method is shown in Figure 11. The risks assessment of an autonomous vehicle was the first crucial step of this study. The second step focused on developing the structure of an online survey to seek information to revise the failure probabilities collected from the literature review and utilized in the first step. The survey instruments were prepared, and after the approval from Institutional Review Board (IRB), the survey was released. The last step of this research was to develop the algorithms of autonomous vehicle navigation

in traffic simulation environment. The micro-level traffic simulation could allow development of an autonomous vehicle environment, where vehicles can be driven by themselves without any human intervention. The autonomous vehicle driving behavior was simulated to estimate the impact of sudden incidents in autonomous navigation due to the risks identified in the first step of this research. The simulation results represent real-world crash scenarios due to the failures of autonomous navigation and their impact on other road users in a mixed traffic stream.



*Figure 11.* Overall research methodology

## **Risk Analysis**

The risk assessment of an autonomous vehicle was divided into four sub-tasks; they are: i) risk identification, ii) risk estimation, and iii) risk hierarchization. After completing these sub-tasks, the risks were estimated using the fault tree analysis method. They were then validated by comparison with real-world data.

**Risk identification.** The first risk identification sub-task consists of a thorough literature review. This task was conducted by performing an extensive review of published reports and peer-reviewed conference and journal papers. This task also led to identifying the sources of potential risks and how much of an impact each potential failure has on the vehicle system as a whole. According to (D. White, 1995), four types of information about an autonomous vehicle system and its components are required for the risk analysis of autonomous vehicles. These four types of information are:

- (a) Nature and characteristics of the failure sources,
- (b) Chain of events,
- (c) Pathways and processes that connect the cause to the effect, and
- (d) Relationship between risk sources and effects.

**Risk estimation.** The next sub-task is risk estimation which can be performed with various analysis methods. Although this study utilizes the fault tree analysis method, other methods were discussed briefly in Chapter 2. After identifying the hierarchical and logical relationships between the identified events in the previous sub-task, the fault trees were developed to determine failure events. The fault tree was started with a top level-event, “autonomous vehicle failure,” and then divided into primary events that could lead to vehicle failure. Then, these primary events were further split into the events that could lead to the failure of the primary events. Here, gate selection between the “And” and “OR” gate plays an important role, because each gate represents a transition from a broad failure to a more localized failure. This process continues to the breakdown of lower level events until none of the events could be broken down any further, and the lowest level

events were classified as basic events. The steps for developing the fault tree are shown in Figure 12.

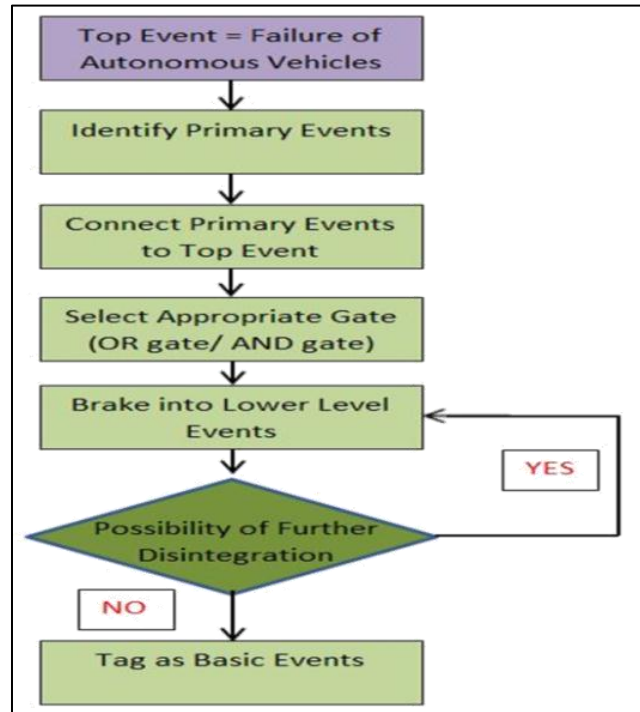


Figure 12. Steps for developing fault tree

**Risk hierarchization.** Along with determining the overall system failure probabilities, the fault tree analysis method allows users to identify the shortest routes, called cut-sets, which can lead to failure of the system within the tree. Each cut-set or path can be obtained directly from the hierarchical relationships of the fault tree. In this study, the identified risks were ranked based on their failure probabilities.

As mentioned in Chapter 2, inputs of risk analysis could be plans, processes and reactions, substances, probability and frequency data, policy and management, environment, text, and historical knowledge. Just as quantitative risk analysis of

autonomous vehicles was set apart as a primary concern here, so probability rate and frequency are also considered important in this study. The ranking of failure risks could help develop cost effective risk minimization strategies, so hierarchization was selected as one of the four different types of outputs of general risk analysis. An analysis on equipment and restricted parts of autonomous vehicles was required here, so their failure probabilities were converted into numerical values. As a result, the probabilistic approach was selected. Figure 13 summarizes the risk assessment sub-tasks.

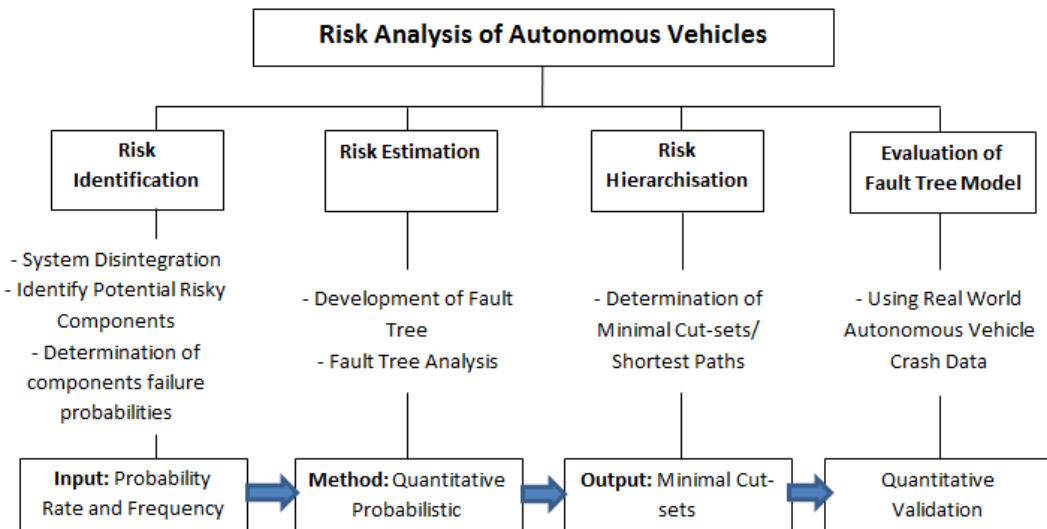


Figure 13. Step by step methodology of risk analysis

**Evaluation of fault tree model.** A fault tree analysis model can be validated qualitatively and quantitatively. The qualitative validation method considers the basic event identification and their relationship with top-level event(s). The quantitative method reviews and measures the failure probabilities (Tupper, Chowdhury, & Sharp, 2014). Finally, the risks estimated in the previous simulated steps were validated by their

comparison with real-world data. For validation, the real-world data available from the California DMV autonomous vehicle testing records were utilized in this study. In Chapter 4, details about three autonomous vehicle risk analysis phases are adopted and their outcomes are discussed.

### **Online Survey**

An online survey was conducted to revise and update the failure probabilities collected from the literature review. The main goal of this survey was to interview the subject matter experts (SMEs) or domain experts to seek further information, which can justify the literature review. The Delphi survey method was used to conduct this online survey.

The Delphi survey method consists of a multi-round interactive anonymous interaction through the questionnaires among participants. The purpose of having multiple rounds is that the participants will review other experts' responses and based on those, revise their previous answers in the following rounds. In this way the responses could be guided to achieve an expected level of consensus in multiple rounds.

**Developing survey instruments.** The causal factors responsible for the overall autonomous vehicle failure were divided into two categories, as previously mentioned in Chapter 3. The first category identifies failure scenarios due to vehicular components, and the second one focuses on the transportation infrastructure components. To collect the intelligences of these two categories, this survey investigates the following two questions:

Research Question 1: Which vehicular component failure would lead to overall autonomous vehicle system failures, and what would be the probability of these component failures?

Research Question 2: What type of transportation infrastructure component failures would cause autonomous vehicles failure? What would be the failure probabilities of infrastructure components?

The questionnaire guided by the Delphi survey method is treated as a medium of anonymous communication among experts from different sectors, where the expressed responses are shared without participant identification. The first-round questionnaire is an open-ended solicitation of ideas. In the following round, the questions are crafted to guide the experts toward an expected level of consensus. To reduce bias, the sequence of questions was randomly generated in different versions. The inputs of experts in this research helped me gather the additional information on causal factors as well as the relationship between causal factors and the impact on overall system success rates. However, these experts were expected to have different perspectives, which meant viewing the autonomous vehicles from different dimensions. Thus, it was evident from the beginning that it would be impossible to obtain a reasonable degree of consensus without separating the experts into different panels. In this study, the experts were divided into three different panels based on the nature of their work area:

- i) Automotive company developers: (for example but not limited to: Google, Uber, Tesla and General Motors),

- ii) University researchers (for example but not limited to: Stanford University, Carnegie Mellon University, University of Michigan, Massachusetts Institute of Technology, and University of Texas at Austin), and
- iii) Component company personnel (for example but not limited to: Velodyne LIDAR, Sanborn LIDAR, and US RADAR Inc.),

Because the Delphi survey population requirements are modest, each panel contained 10 to 18 members, who are experts in the focus area of study (Hasson, Keeney, & McKenna, 2000). To enable global perspectives, 20% of the invited experts on each panel were chosen from outside the United States.

The Kendall's  $W$  coefficient of concordance, a statistical test, was utilized to measure the level of consensus. In this test, a high value of  $W$  ( $> 0.8$ ) means that the participants applied essentially the same standard in judging the probabilities of the vehicular components or transportation infrastructure components. The steps involved in this survey are summarized in Figure 14.

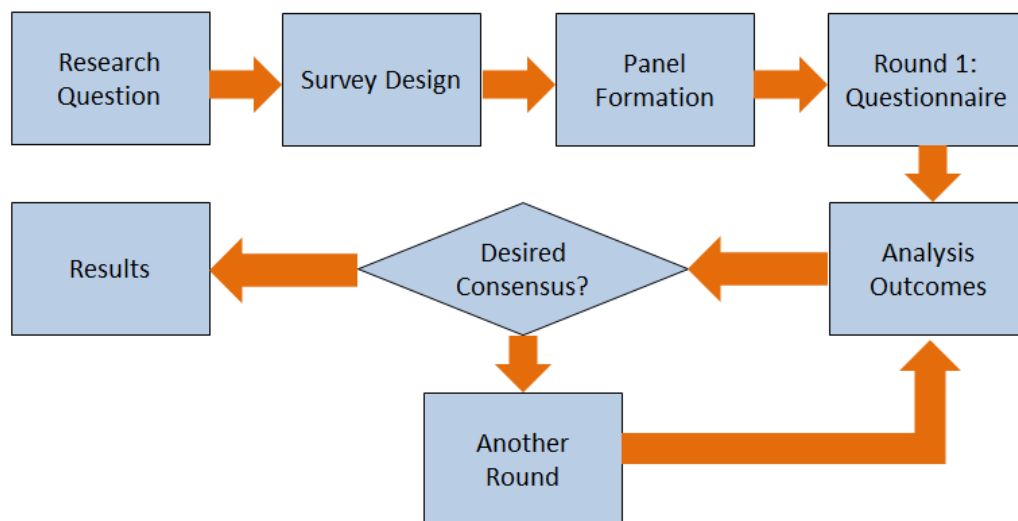


Figure 14. Steps involved in online survey



## **Simulation of Autonomous Vehicle**

The fault tree analysis results estimate the failure probability of autonomous vehicles sharing the roads with human drivers and other road-users. However, it is essential to further study how these failure probabilities will impact the overall efficiency of the roadway infrastructures. Keeping this object in mind, micro-level traffic simulation with an autonomous navigation environment was developed and analyzed using the fault tree results in simulation. The autonomous driving behavior was modeled in traffic simulation and the probabilities of autonomous vehicle failure were integrated into the model. In this research, vehicle platooning was evaluated as autonomous navigation; however autonomous vehicle lane changing behaviors were not included at this stage. After modeling the simulations, the results of traffic microsimulation were imported into a conflict analysis tool, which could estimate the reduction in crash frequencies. Finally, the impacts of autonomous navigation failures were studied using simulation analysis.

The modeling of autonomous vehicles was split into four sub-tasks:

- i) traffic network modeling,
- ii) formulation of algorithms,
- iii) modeling multiple scenarios, and
- iv) conflict analysis.

It is important to mention that the traffic microsimulation software package, PTV Vissim (version 7.00- 32 bits), was used in this study to model the road networks ("VISSIM 7 User Manual," 2015). The benefit of using this simulation platform is that it enables users to develop and simulate specific user-defined driving behavior for either a specific percentage of total vehicles or all vehicles. Additionally, the Surrogate Safety

Assessment Model (SSAM), a conflict analysis tool, was utilized in this study (Pu & Joshi, 2008).

**Traffic network modeling.** A segment of Interstate-476 in Pennsylvania was selected as the study region, and its traffic network was modeled in Vissim. The study area was bounded by US Route 3 near Haverford, PA on the north to I-95 near Woodlyn, PA on the south. The geometric parameters (for example but not limited to: number of lanes, lane widths, and turning radius) and designs were collected and modeled based on Google Maps using the satellite feature. Figure 15 shows the four junction points of the entire study's road network. All related intersections and local road networks are in the simulation model as well as traffic counts collected from the 2015 Pennsylvania Highway Statistics Report, available on the PennDOT Traffic Information website (PennDOT, 2016).



*Figure 15.* Traffic simulation model of I-476 in Pennsylvania developed in Vissim

The Vissim model includes 99 links and 186 connectors; totaling 49.87 kilometers of traffic network. Fourteen traffic signals were modeled on entrance ramps. Additionally, 232 conflict points were coded to represent the merging areas in the study traffic network. The model was then calibrated by adjusting speed distribution, human driving behaviors at merging regions and speed decision points by Hard Should Running Clinic Team.

**Number of simulation runs.** It is important to introduce variability in traffic microsimulation, because even on a specific segment of highway, it is expected that traffic patterns will fluctuate based on multiple parameters, i.e., (for example but not

limited to time of the day, workdays, weather, seasons and traffic crashes). With current computation systems, it is not possible to generate a sequence of random numbers which cannot be reasonably predicted (Bahaaldin, Fries, Bhavsar, & Das, 2017; Vattulainen & Ala-Nissila, 1995). However, in Vissim, a parameter called “random seed” can actually initialize randomness in traffic patterns. Thus, a traffic microsimulation model with the same random seed value can produce similar results for operational parameters (i.e., measure of effectiveness), such as travel time, network speed, and density. If the random seed value is varied, then the built-in stochastic functions in Vissim will generate a stochastic variation of traffic arrivals in the microsimulation. Furthermore, the results generated from multiple runs of a single traffic microsimulation are required to draw a conclusion with statistical validity. However, it is essential to prove that the results are a true representation of the calibrated simulation model and not skewed towards a statistical outlier. The average results of multiple runs using different random seed values should stay within the true average range of the model, i.e., confidence level. It is important to mention that the average results of multiple runs do not necessarily have to be representative of real-world scenarios, unless the model has been pre-calibrated.

In this research, the initial value of the random seed was assigned as 5; then, this value was incremented by 5 in each simulation run. It is recommended that the initial number of simulation runs should be 10 to determine the confidence level of simulated results (ODOT, 2011; WSDOT, 2014). Based on that, the base model was run initially 10 times using different seed values starting from 5 and then increased by 5 after a single run. The average network speed and average travel time values for each travel time measured segment were recorded to calculate the true statistical average. Then a Student’s t-test

was conducted to validate the results collected from 10 initial runs. The t-statistic equation is expressed as:

$$t - statistic = \frac{Estimate - Parameter}{Standard\ Error(Estimate)} \quad (1)$$

This equation can also be written as:

$$t - statistic = \frac{\bar{x} - \mu}{S/\sqrt{N}} \quad (2)$$

where,  $\bar{x}$  = sample average

$\mu$  = population average

$S$  = standard deviation of sample

$N$  = sample size, i.e. number of simulation runs

Furthermore, this equation was rearranged to calculate the number of simulation runs required to achieve the average values of parameters within a predetermined confidence level. Considering a confidence level of 95%, the following equation was developed (WSDOT, 2014):

$$N = \left( 2 \times t_{0.025, DF=N-1} \times \frac{S}{R} \right)^2 \quad (3)$$

where,  $R$  = 95% confidence interval for a true average

$S$  = standard deviation for selected parameters, i.e., measure of effectiveness

$t_{0.025, DF=N-1}$  = Student's t-statistics for two-sided error of 2.5% with  $N - 1$  degrees of freedom

However, in this research, the network average speed values for different time intervals collected from initial 10 simulation runs were used to determine the number of simulation runs required to draw a convincing conclusion. The error tolerance was set at 10%. The number of simulation runs required was calculated using a network speed

average for a time interval between 600 and 1500 seconds as per Equation (3). The network average speed values and detailed calculations are presented in Appendix A.

$$N = \left( 2 \times t_{0.025, DF=9} \times \frac{s}{R} \right)^2 \quad (4)$$

$$N = \left( 2 \times 2.2622 \times \frac{0.8065}{0.10 \times 30.6636} \right)^2 = 1.42 \text{ runs}$$

This calculation shows that 10 simulation runs were enough to achieve the average of parameters within a 95% confidence level, which supported a statistically validated conclusion. However, in this research total 11 simulation runs were executed to easily address the simulation run that provide median values of the assigned parameters.

Furthermore, the average travel time value recorded for each time interval for each measured travel time segment was used to calculate the number of simulation runs required, since travel time was considered the measure of effectiveness for performance evaluation. It was found that 10 simulation runs were enough for reporting results within a 95% confidence level. However, as mentioned earlier, the calibrated simulation model was run 11 times in this research. The detailed calculations are provided in Appendix A.

**Formulation of the autonomous navigation algorithm.** Traffic analysis in computer simulation has become familiar nowadays with the advancement in computing power (Pel, Bliemer, & Hoogendoorn, 2011; Rossetti & Ni, 2010). Researchers found that a traffic simulation model could represent real-world scenarios after proper calibration and validation, and the results of simulation were satisfactory (Gomes, May, & Horowitz, 2004; Mahmassani, Hou, & Dong, 2012). The Vissim (version 7.00), one of the micro-level traffic simulation software platforms, was utilized to model traffic network in this research. Vissim was chosen because of its component object model (COM) interface and its external driver model (EDM) availability to simulate

autonomous driving behavior. However, this software platform has its own in-built driving model based on the Wiedemann algorithm developed in 1974. This driving model was built to predict non-autonomous driving behavior at the micro-level. Therefore, to simulate autonomous navigation, user defined driving algorithms were required. However, Vissim allows building a platform which can integrate the EDM algorithms, coded by the user. In this research autonomous driving algorithms were developed and integrated with the Vissim platform. These external driving algorithms were used to replace the in-built human driving behaviors and simulate the autonomous driving environment. This autonomous driving environment was used to assess the impact of risk analysis results at the micro-level. Furthermore, this autonomous vehicle simulation was applied to predict future traffic scenarios when autonomous vehicles would be implemented in roadways. It is important to mention again that autonomous platooning was considered in this research, where driving maneuvers like lane change were not considered.

***External driver model algorithm.*** The external driver model will allow replacing the internal driving behavior and implementing user-defined behaviors. Based on user-defined algorithms, a dynamic link library (DLL) written in C/C++ is integrated with the simulation model and was activated during the simulation run (code is presented in Appendix B). In every single simulation time step, Vissim calls the DLL code to determine the status of the specific vehicle in the next simulation time step ("VISSIM 7 User Manual," 2015).

The steps followed to develop and run the external driver model are described here. At first, a new vehicle type was created in Vissim, and this vehicle type followed

the autonomous driving algorithms to move one position to another position. To integrate the autonomous driving algorithms, a dynamic link library (DLL) was created in C++ language. The DLL files are comparable with the EXE files; however, they are not directly executable like EXEs. The DLL files require a platform/ program to execute, and this creates interdependency. Similarly, the DLL file developed in this study was executed in the Vissim environment, where Vissim communicates with the DLL file to predict the next move of a specific vehicle type, i.e., in this case, autonomous vehicles. This DLL file for autonomous navigation has three parts; they are:

- i) Main function
- ii) Header File
- iii) Resource File

The main function encompassed the algorithms of driving behaviors, and the header file was used to translate the outcomes of algorithms into Vissim variables. Finally, a resource file was developed to create the sequence of functions needed to execute while running the traffic simulation. The main file contains three functions required to move autonomous vehicles, they are:

- (i) Set value: Vissim passes current information of the vehicle,
- (ii) Get value: retrieve new information based on defined algorithms, and
- (iii) Execute command: Passes the request of execution to Vissim.

The overall flow of information is presented in Figure 16. Based on the current vehicle information, algorithms identify the leading vehicle type and estimate the speed of the leading vehicle at the time. Then a polar question arises as to whether the leading



vehicle is a similar autonomous vehicle type. If the answer to this question is yes, then algorithms estimate the distance between the current vehicle and leading vehicle.

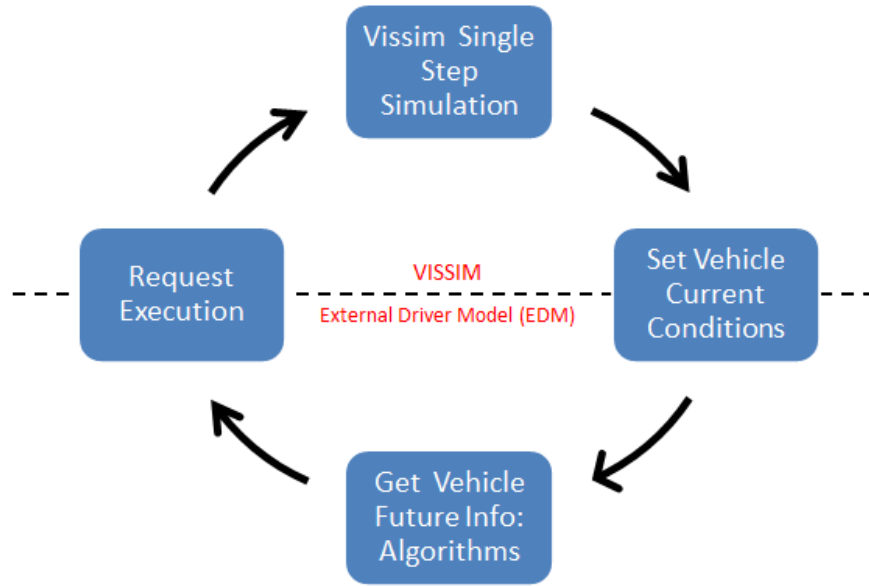


Figure 16. Flow of information between Vissim and EDM

The distance between the current and leading vehicle is used as parameter for creating a platoon. The threshold values for vehicle platooning are 6.6 feet as the desired gap or distance and 3.3 feet as the emergency gap distance. If a vehicle is more than 6.6 feet from the leading vehicle, then the current vehicle will accelerate to get closer to the leading vehicle. Hence, if the current vehicle comes within less than 3.3 feet from the lead vehicle, the current vehicle will decelerate to increase the gap between them. The architecture of autonomous driving is given in Figure 17. Additionally, the mathematical formulation of the external driver model is explained below.

Mathematical Formulation:

Speed Difference,

$$S_{diff}(t) = v_{ego}(t) - v_{leading}(t) = \text{velocity of ego vehicle} - \text{leading vehicle velocity} \quad (6)$$

Case 1: If  $S_{diff}(t) > 0$  &  $gap(t) > gap_{des}$ , then

$$a(t) = \frac{-(S_{diff}(t))^2}{2 \times (gap(t) - gap_{des})} \quad (7)$$

where,  $a(t)$  = desired acceleration

$gap(t)$  = gap between the ego vehicle and leading vehicle

$gap_{des}$  = desired gap

Case 2: If  $S_{diff}(t) > 0$  &  $gap_{em} < gap(t) < gap_{des}$ , then

$$a(t) = \frac{-S_{diff}(t)}{t - (t-1)} \quad (8)$$

where,  $gap_{em}$  = emergency gap distance

Case 3: If  $S_{diff}(t) > 0$  &  $gap_{em} > gap(t)$

$$a(t) = \frac{-(S_{diff}(t))^2}{2 \times (gap_{em} - gap(t))} \quad (9)$$

Case 4: If  $S_{diff}(t) < 0$  &  $gap(t) > gap_{des}$ , then

$$a(t) = \frac{(S_{diff}(t))^2}{2 \times (gap(t) - gap_{des})} \quad (10)$$

Case 5: If  $S_{diff}(t) < 0$  &  $gap_{em} < gap(t) < gap_{des}$ , then

$$a(t) = \frac{S_{diff}(t)}{t - (t-1)} \quad (11)$$

Case 6: If  $S_{diff}(t) < 0$  &  $gap_{em} > gap(t)$ , then

$$a(t) = \frac{(S_{diff}(t))^2}{2 \times (gap_{em} - gap(t))} \quad (12)$$

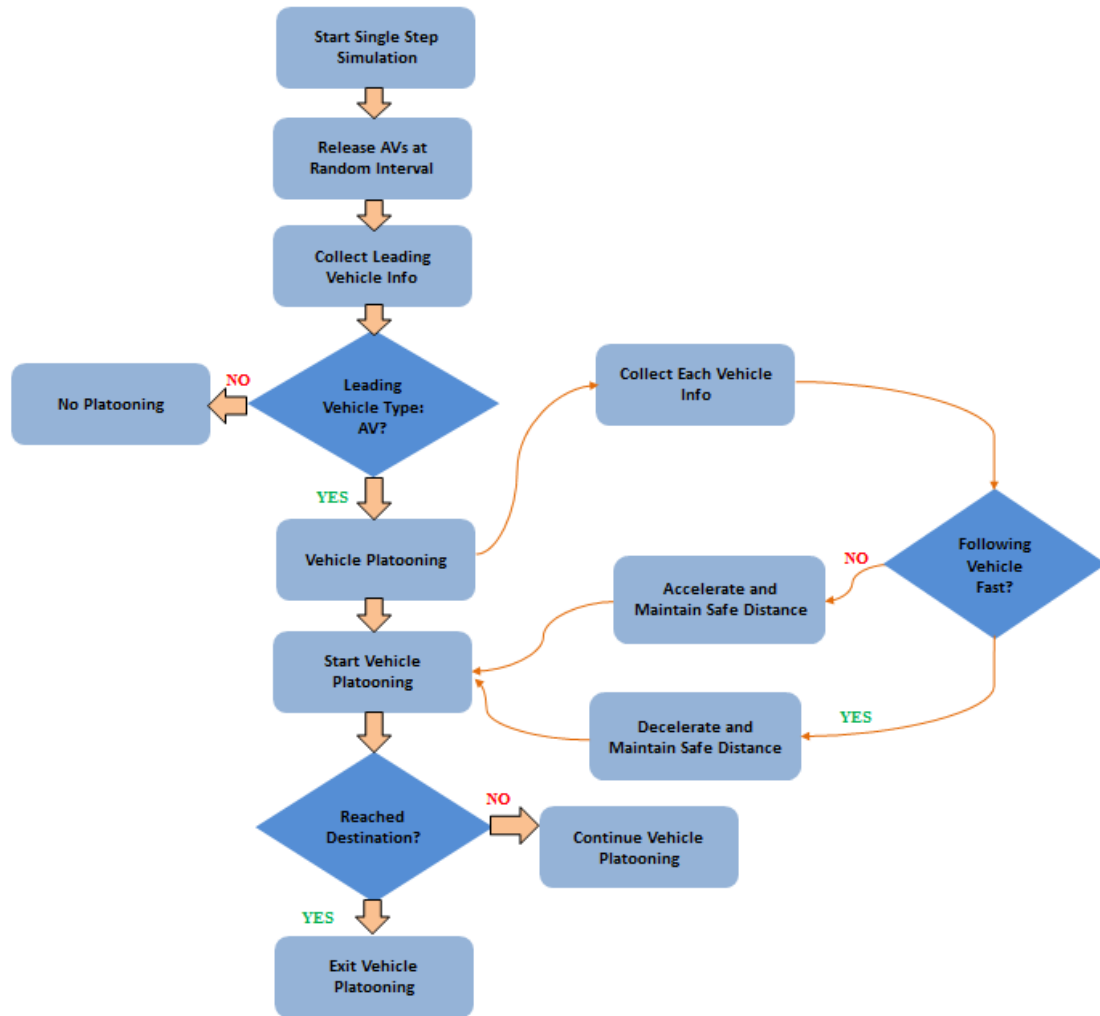


Figure 17. External driver model algorithm

**Modeling multiple scenarios.** The base model was coded with autonomous vehicles' market penetration level with 0 percentages representing current mixed traffic scenarios. This percentage of market penetration level then gradually increased to simulate future scenarios. For example: 10%, 25%, 50% and 90%. It is important to mention that autonomous passenger cars are considered in this risk analysis research, and other different transportation modes not considered for example but not limited to transit, heavy-goods vehicles, and motorcycles. The Visssim vehicle types represent vehicles

other than passenger cars, i.e., transit vehicles, trucks, motorcycles, etc., that were not modified at all. They were used in Vissim vehicle navigation algorithms built into the software platform. Furthermore, the demand for these vehicles did not change over time; and hence their penetration level was constant over the time period.

However, demand analysis of autonomous vehicles was not included in this research, since it is not within the scope of the research. It was assumed a gradual increase of these vehicles will continue over a period of time, i.e., years. Furthermore, researchers have predicted that the autonomous vehicles could increase the travel distance, i.e., vehicle mileage, and hence, congestion will also increase as vehicle travel becomes more convenient (Smith, 2012). In (Stefan Trommer et al., 2016), Trommer et al. estimated that vehicle travel distance by 2035 will see an additional increase of at least 3 to 9% after autonomous vehicles are implemented on the road. Additionally, disabled persons, elders, and children, who were restricted from driving altogether, will have their independent mobility. However, these new road user groups, i.e., disabled persons and elders, may increase the number of vehicles waiting behind the “red” traffic signal light by up to 11% (Michael Sivak & Schoettle, 2015). It is important to include these perspectives in simulation modeling, since network travel times could deviate due to their impact. However, these futuristic problems are not within the scope this research and their impacts have not been validated yet using real world data.

**Conflict analysis.** The objectives of conflict analysis are to identify the improvements in traffic crash reduction after implementing autonomous vehicles on our roadways to quantify the impacts of autonomous vehicle crashes on the overall performance of transportation infrastructure. This analysis could be a platform where

results of the fault tree models could be integrated with traffic simulation modeling. Based on the objectives, the conflict analysis could be divided into two phases.

***Crash frequency estimation.*** Using simulation modeling, the assessment of traffic safety, i.e., crash analysis, is always difficult because of pre-built evasive algorithms in traffic simulation software. However, researchers have developed effective analyses of the relationships between traffic crashes and traffic conflicts (F. Amundsen & Hyden, 1977), where the intersection of two or more vehicles is defined as a conflict. Until recently traffic conflicts were surveyed by trained personnel by observing a traffic fleet. But this method could be questionable due to the surveyor's subjective judgements (Huang, Liu, Yu, & Wang, 2013). After a thorough research, the "Surrogate Safety Assessment Model (SSAM)" was developed by a research team at Siemens ITS, sponsored by the Federal Highway Administration (FHWA), to integrate traffic simulation modeling and conflict analysis together. In this tool, a crash is considered between two vehicles, which are on a collision course, but due to evasive actions the crash is prevented. This model uses the trajectory files imported from microscopic traffic simulation models and calculates the number of conflicts utilizing several algorithms. The number of conflicts, types of conflicts, severity and location of conflicts are the outputs of SSAM models. There are five parameters used in this model to estimate the severity of simulated conflicts: time-to-collision (TTC), post encroachment time (PET), deceleration rate (DR), maximum speed (MaxS) and speed difference (DeltaS). Three different types of crashes can be modeled using the SSAM tool. These crash types are separated based on the conflict angles between the vehicles. Figure 18, from the SSAM manual, shows the angle variation of these crash types (Pu & Joshi, 2008). Three types of

crash are considered in SSAM: 1) rear end collisions, 2) lane-changing conflicts, and 3) crossing collisions. However, traffic crashes are certain, where conflicts are more frequent than crashes. It is really important that the SSAM models are calibrated and validated using real-world data to estimate realistic crash frequency values (Vasconcelos, Neto, Seco, & Silva, 2014).

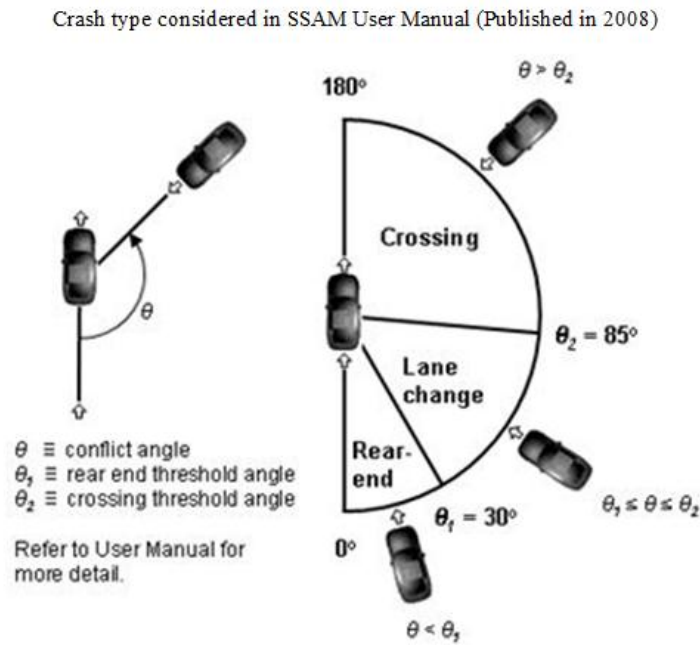


Figure 18. Three types of Crash in SSAM (Pu & Joshi, 2008)

In this research, the SSAM tool was used to estimate and compare crash frequencies between two traffic models, i.e., specifically simulation models with different autonomous vehicle market penetration levels. To conduct conflict analysis, trajectories files were generated in the Vissim model after first running simulations. These trajectories files with the “.TRJ” extension were originally a binary file that contained the course of vehicle positions, i.e. trajectory, through the modeled traffic network. These

trajectory files were imported to the SSAM model to estimate the frequency of traffic conflicts. However, since the traffic simulation model was calibrated and validated using real-world data the requirements of SSAM model calibration were overridden. Figure 19 represents the integration platform of traffic simulation software and the conflict analysis tool.

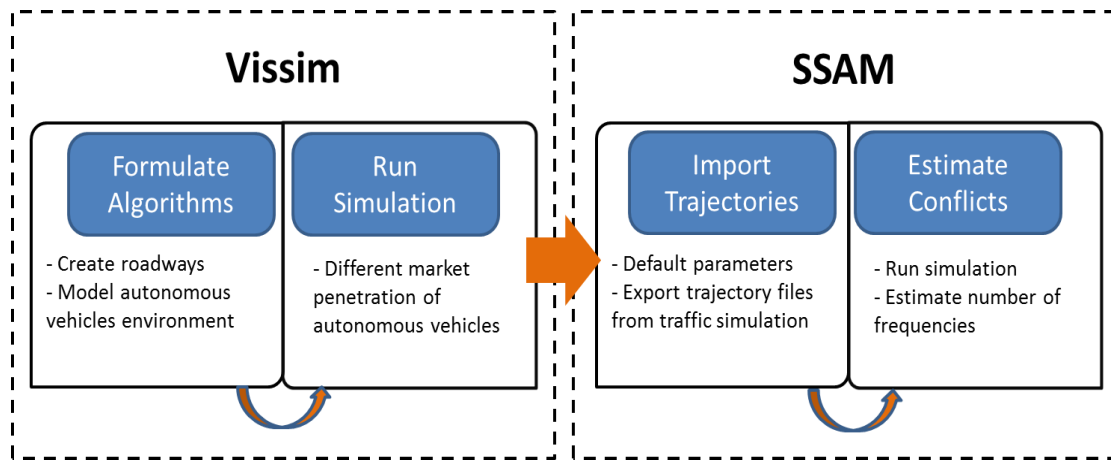


Figure 19. Integration platform of Vissim and SSAM

In SSAM software, the default values of TTC and PET are 1.50 and 4.00 seconds, respectively. These values were estimated based on previous research on urban signalized intersections, i.e., low-speed road networks (25 to 30 mph). However, it is expected that the perception reaction time (PRT) and maneuver time (MT) will be lower for autonomous vehicles than for non-autonomous drivers. As a result, along with the default TCC value, two other values, 0.9 and 1.2, were used in this research to investigate the variation in the conflict frequencies based on TTC values. A PET value of 3.00 was additionally examined besides the default PET value.

***Integration of fault tree and simulation modeling.*** Risk analysis of an autonomous vehicle was conducted to estimate the failure probability of autonomous navigation due to either vehicular components or transportation infrastructure component failures. This failure probability represents the number of incident failures that could occur per certain distance traveled over the period of a vehicle's life. In other words, the possibility of a traffic crash will be high after certain distances of travel, i.e., in this research per 1,000,000 miles. Later on, the results of risk analysis models were integrated in the Vissim model to estimate the impacts of these failures on the performance of transportation infrastructures. However, the years a vehicle is driven before it dies depends on various parameters, i.e., vehicle maintenance, annual mileage, and weather. On an average, it is expected that the life of a new vehicle should be around 8 years. Modeling the entire life cycle of a vehicle is not feasible in traffic simulation, not even for one vehicle driving 1,000,000 miles as determined in the network modeled earlier. Instead, it was assumed that all vehicles released in simulation would travel 1,000,000 miles collectively; then, a traffic crash scenario would arise. The overall algorithm is represented in Figure 20. A visual basic code was utilized to generate the failure of autonomous navigation, which is presented in Appendix C.



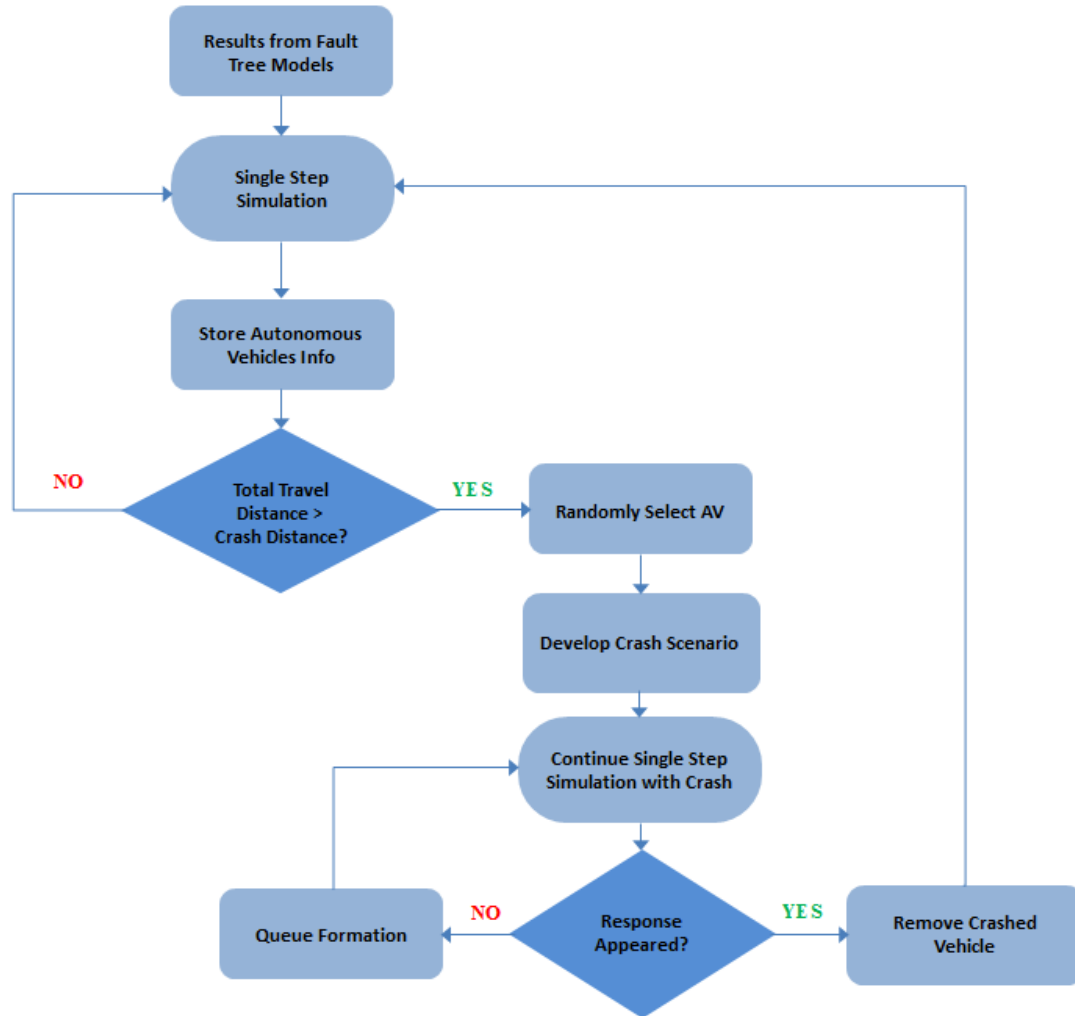


Figure 20. Integration platform of fault tree and traffic simulation model

## Summary

In summary, the three major steps, i.e. overall research method, in this research process are described in this chapter. The steps for fault tree-based risk analysis are mentioned at the beginning of this chapter, followed by the survey structure. The chapter also explains the details of traffic microsimulation development. The analysis and results of the fault tree-based risk analysis are presented in the next chapter.

## **Chapter 4**

### **Risk Analysis of Autonomous Vehicle**

The comprehensive risk analysis of autonomous vehicles in a mixed traffic stream is presented in this chapter, which can be divided into four interconnected sub-sections. Because investigating vehicular components and analyzing their behaviors are the first crucial step of risk analysis, this chapter starts with a detailed description of the autonomous vehicle risk identification process. The next sub-section summarizes the risk estimation, followed by risk hierarchization. The validation of risk estimation is presented in last sub-section of this chapter.

#### **Risk Identification**

Autonomous vehicles are equipped with various sensors and actuators, and communication platforms, which are interconnected to sense the roadway and other road users. They comply with traffic rules and regulation and navigate in the traffic stream without human intervention. Each of these components has its own failure mechanisms and reliability functions. Investigating these failure mechanisms is required to ensure safe navigation. To identify and analyze the basic components, risk identification was started by disintegrating the autonomous vehicle system into each of its individual components, and then analyzing their behavior. A detailed literature review of published reports, peer-reviewed conference and journal papers, and other published materials was conducted to estimate the failure probability of each component and develop hierarchical and logical relationships between the top-level event (failure of an autonomous vehicle) and different autonomous vehicle components.

We have been seeing that the transition from conventional system to advanced technologies normally takes place over a period of time such as the quality and extent of computer upgrades (new models) from the 1960s until now. Therefore, it is expected that the transition from a conventional non-autonomous vehicle fleet to an autonomous vehicle fleet will likely go through a series of gradual changes over the years. This suggests that autonomous vehicles will share the roadway with conventional vehicles such as cars, transit buses, trucks, as well as bicycle riders, motorcyclists, and pedestrians for many years to come. As a result, a risk analysis of autonomous vehicles needs not only to include the failure mechanisms of vehicular components, but also consider the impacts of transportation infrastructure component failures.

The risk identification process was divided into two subcategories to estimate failure risks of autonomous vehicles due to different vehicular components and transportation infrastructure components. The first category focused on identifying threats from autonomous vehicular components, and the second category focused on identifying threats from infrastructure components, including threats from other non-autonomous vehicles.

**Autonomous vehicle components.** In Chapter 2, the literature review presented automotive features which could convert a conventional vehicle into an autonomous vehicle. These automotive features then led to the development of the necessary sensors and components of an autonomous vehicle. All these sensors and components were categorized into four major subsystems: hardware, software, communication, and human-machine interface. The hardware system includes sensors and components, such as LIDAR, radar, camera, GPS, wheel encoders, and the integration platform. The sensors in

hardware system are utilized to collect the surrounding information, whereas the software subsystem consists of the data processing software required for autonomous navigation. Vehicle-to-vehicle (V2V) and vehicle to infrastructure (V2X) communication platforms are included in the communication subsystem, along with communication database failure. The final major subsystem is the human machine interface, which is used as a personal assistant system that filters the human voice for commands to control various autonomous driving functions. It is important to note that in this study; only additional new technologies that convert a conventional human operated vehicle into an autonomous vehicle were considered.

LIDAR, the primary technology being used for autonomous navigation, can fail owing to several reasons, including laser malfunction, mirror motor malfunction, optical receiver damages and electrical failures (Duran et al., 2013b). Similarly, camera vision is another very important component on an autonomous vehicle, capable of providing physical information about surroundings (for example but not limited to: obstacles, road signs, and pedestrians). This system can also fail; however, misalignment, a missing filter, dirty or damaged lens, and even improper lighting are only a few problems than can lead to the failure of a camera. Detection failure of radar was estimated and mathematically modeled so that the detection could fail two times out of 100 runs. After real-world testing, it was estimated that the GPS system could fail due to variations in the signal environment. Additionally, a wheel encoder could fail due to the loss of motor stator synchronization and rotor positions. Furthermore, the integration platform is used for communicating between all the sensors and units; thus, the hardware sensors communicate with the data processing unit and the software unit, and any platform failure

could be critical to the continuance of a vehicle's autonomous navigation. The failure probability of the integration platform was 2% when a two-state model was developed.

Since the driving responsibilities are essentially shifting away from active human control to complete automation, the reliability of an autonomous vehicle software system needs to be validated before deploying these vehicles on the roads. In an experiment, it was found that software failed to generate a signal 1% of the time based on the array definition language (ADL) statements. However, the database server could lose its functionality due to operability and connectivity failures. In addition, the human machine interaction platform could play an important role in the performance of the autonomous vehicle. The National Aeronautics and Space Administration (NASA) analyzed a dataset of over 115 months and calculated the probability of human error (i.e., wrong commands) over certain periods of time. Another study was conducted to estimate the rate of system failures in detecting human commands and found that the detection could fail 1.4 times out of 100 human commands.

Additionally, during location updates, long-term evolution (LTE) networks could fail 5.88% times due to its control-plane failures. Other researchers evaluated the Wi-Fi reliability with 10 vehicles, where messages were transmitted to and from moving vehicles using open Wi-Fi. However, due to the high rate of package losses these transmissions failed 5.125% times over the experiments. Besides these, database service has a failure probability of 3.86% due to connectivity losses and operability failures. The failure probabilities for all these components along with reasons for failure are summarized in Table 4 based on findings from literature reviews. Furthermore, the failure of the vehicle's mechanical system was not in the scope of this study as it is not a part of

the system that converts a conventional vehicle into an autonomous vehicle.

Table 4

*Failure probabilities of autonomous vehicular components*

Basic Events	Description	Methods	Experiment Type	Failure Probability (%)
LIDAR failure	Laser malfunction, mirror motor malfunction, position encoder failure, overvoltage, short-circuit, optical receiver damages.	Bayesian belief network	Simulation	10.0000% (Duran et al., 2013b)
Radar failure	Detection curves drawn with respect to signal and noise ratios	Chi-square distribution	Mathematical modeling	2.0000% (Swerling, 1997)
Camera failure	Foreign particles, shockwave, overvoltage, short-circuit, vibration from rough terrain, etc.	Bayesian belief network	Simulation	4.9500% (Duran et al., 2013b)
Software failure	System had to generate outputs from array definition language (ADL) statements	Extended Markov Bayesian network	Experiment (3000 runs)	1.0000% (Bai, 2005)
Wheel encoder failure	Encoder feedback unable to be transferred, which can cause loss of synchronization of motor stator and rotor positions	Kalman filter	Experiment	4.0000% (Goel et al., 2000)
GPS failure	Real-life tests performed with high sensitivity GPS in different signal environments (static and dynamic) for more than 14 hours	Least squares	Experiment (at 4 different locations)	0.9250% (Kuusniemi)
Database service failure	Using new empirical approach, connectivity and operability data of a server system was collected	Generic Quorum-system Evaluator (GQE)	Experiment (for 191 days)	3.8600% (Amir & Wool, 1996)
Communication failure	Wi-Fi: Periodic transmission of 1000-byte frames (average conditional probability of success after previous success considered)	In IEEE 802.11b network	Experiment (with 10 vehicles)	5.1250% (Eriksson, Balakrishnan, & Madden, 2008)

Table 4 (continued)

Basic Events	Description	Methods	Experiment Type	Failure Probability (%)
Communication failure	LTE: Network unavailability during location update in mobility was considered here	Application of CAP theorem	Experiment	5.8800% (Li, Yuan, Peng, & Lu, 2016)
Integrated platform failure	A two-state model with failure rates was developed to estimate the computer system availability	Markov chain model	Mathematical modeling	2.0000% (Goyal, Lavenberg, & Trivedi, 1987)
Human command error	Three datasets of over 115 months from NASA was analyzed and then validated by three methods (THERP, CREAM, and NARA) to facilitate NASA risk assessment	Human Reliability Analysis	Experiment (from December 1998 to June 2008)	0.0530% (Faith Chandler et al., 2010)
System failed to detect human command	System unable to detect the accurate acoustic command; Driver inputs the wrong command, and system unable to detect wrong commands	Artificial neural networks (ANNs) on clean speech	Experiments (37 subjects: 185 recording)	1.4000% (Dupont & Luettin, 2000)

**Transportation infrastructure components.** Autonomous vehicles are expected to be gradually introduced to general traffic with initially low market penetration rates. Thus, the surrounding infrastructure of an autonomous vehicle including other non-autonomous vehicles (i.e., human drivers) can have a tremendous impact on autonomous navigation. Failure will create a reliability issue for the autonomous vehicle. Recent reports of autonomous vehicles testing submitted by companies that conduct autonomous vehicle testing, indicate that the majority of autonomous vehicle-involved crashes are due to human drivers sharing the road with autonomous vehicles (Delphi, 2016; Google, 2016; Mercedes-Benz, 2016; Nissan, 2016; Volkswagen, 2016).

Autonomous vehicles have been tested in mixed traffic streams during low market penetration to determine their level of performance, and non-autonomous vehicle drivers are a major issue in mixed traffic streams. Thus, crash records related to reckless driving, distraction, vehicle breakdown and fatigue were collected from traffic crash reports involving non-autonomous vehicles of the Virginia Department of Transportation (VDOT) and New York State Department of Transportation (NYSDOT) (NYSDOT, 2015; VDOT, 2015). The data were then converted into crash rate per mile of autonomous vehicles (i.e., basic events' failure probability) in the fault tree. In this research, the market penetration rate of 10% of the autonomous vehicles was used to calculate the failure probability of an autonomous vehicle traveling in a mixed traffic stream. To consider the worst-case scenario, 10% of total crashes on a roadway are considered to affect autonomous vehicle's navigation in a mixed traffic stream. A sample calculation is presented in Appendix D to describe the details of failure probability calculation for an autonomous vehicle (AV), when it is involved in a crash due to reckless driving, fatigue or distraction of a non-autonomous vehicle (non-AV) driver.

Incident rates due to poor weather and road conditions were collected from VDOT and NYSDOT as traffic crashes attributed to bad/poor road conditions were considered as transportation infrastructure failures. Bicyclists and pedestrians involved in crashes were also included. A study in Hawaii found that 83.5% crashes between motor vehicles and cyclists were caused by motorists and the other 16.5% were caused by cyclists (Schroeder & Wilbur, 2013). Weather is a huge deterrent to autonomous vehicles, especially since few autonomous vehicles have been tested in adverse weather. Construction work zones crashes were also considered; particularly rear-end crashes



(Ullman, Finley, Bryden, Srinivasan, & Council, 2008). Table 5 reports failure probabilities of these infrastructure components, as reported in the literature.

Table 5

*Failure probabilities of basic transportation system infrastructure components*

Basic Events	Description	No. of Crashes	Failure Probability (% per Mile)	References
Non-autonomous vehicle crashes	Crashes due to reckless driving, fatigue, hardware and distractions	133,901 (per 100 million miles)	0.0134%	(NYSDOT, 2015; VDOT, 2015)
Cyclists	9 million daily bike trips with cyclists responsible for crashes	3,090	$4.0897 \times 10^{-6}$ %	(NHTSA, 2015; Santos, McGuckin, Nakamoto, Gray, & Liss, 2011; Schroeder & Wilbur, 2013)
Pedestrians	Crashes where pedestrians at fault during annual 42 billion walks	8,625	$2.9337 \times 10^{-6}$ %	(J. Richard Kuzmyak & Dill, 2012; NHTSA, 2016c; Santos et al., 2011; Schroeder & Wilbur, 2013)
Construction zones	Among all work zones 41.33 percent were rear-ended crash	36,208	$7.6264 \times 10^{-6}$ %	(FHWA, 2015; Ullman et al., 2008)
Weather related incidents	Adverse weather: fog, mist, rain, severe crosswind, sleet, snow, dust/ smoke	22,375 (per 100 million miles)	0.0022%	(VDOT, 2015)
Road conditions	Crashes related to improper lane marking and pavements conditions	656 (per 100 million miles)	$6.5600 \times 10^{-5}$ %	(NYSDOT, 2015)

## **Risk Estimation**

After estimation of the failure probabilities of vehicular components and transportation infrastructure components, the next step of autonomous vehicle risk analysis is calculating the top-level failure rates. According to Stanford University's Department of Global Ecology, "Risk assessment often begins by looking at one part of the problem, usually the source of the effect, rather than considering the system as a whole" (A. V. White & Burton, 1980). Fault-tree analysis approaches assessment from a top-down approach, as risk estimation begins with the root cause of the basic/ primary components failures and proceeds to estimate the failure probability of the top-level event. Furthermore, this method can provide the shortest path to reach that top-level failure from a single component failure. Because of these benefits, the fault tree analysis model was utilized to perform risk estimation in this study. The previous task was risk identification guided to analyze the behavior of vehicular components and transportation components in mixed traffic streams, and to estimate the failure probabilities of these components. Based on these failure probabilities, fault-tree models were developed and will be explained in following subsections. The risks associated with autonomous vehicles were categorized into two sub-sections, vehicular components and transportation infrastructure components; thus, two separate fault tree models were developed based on the risks introduced in the two following sub-sections. The two fault trees models are:

- (i) Fault tree model for autonomous vehicle failure due to vehicular component failures, and
- (ii) Fault tree model for autonomous vehicle failure due to transportation infrastructure component failures.

However, these models were eventually combined to estimate the overall risk of failure, i.e., failure of an autonomous vehicle in mixed traffic streams.

**Fault tree for autonomous vehicular component failures.** The fault tree is developed by disintegrating an overall system into lower resolution events. This process continues until no further disintegration can take place. These terminating events are called “basic events”. The failure of the overall system is referred to as a “top-level event” and the events linking a top-level event with its basic events are called “intermediate/ casual events.” The top-level event and its basic events are interconnected based on hierarchical and logical relationships between the events that led to failure of the top event. In a graphical representation of a fault tree, these logical relationships are presented as “gates.” The “AND” and “OR” gates are widely used to illustrate the relationship between input and output events. Risk estimation quantifies the failure rate of the top-level event and is represented as a percentage in decimal format. This estimation takes all basic events into account and determines the failure rate based on Boolean algebra. The algebraic equations that are performed are determined by the gates used and the statistical model that was used when inputting basic events.

The first fault-tree model was developed considering the failure of an autonomous vehicle due to vehicular components. The Isograph FaultTree+ software, which allows various statistical models to model basic event failure probability distribution, was used for fault the tree analysis ("Commercial Software for Fault Tree Analysis,"). For this study, a “fixed probability” statistical model was used to perform the risk analysis ("Commercial Software for Fault Tree Analysis,"). After allocating basic event failure probabilities and solving the fault tree, a failure rate of 14.22% was determined for the

autonomous vehicle due to its components' failure, which means that autonomous vehicle operations could fail 14.22 times over its lifetime due to component failure. Figure 21 illustrates the fault tree with failure probabilities including only autonomous vehicle components.

**Fault tree for transportation infrastructure component failures.** Following the same steps applied in first fault tree, the second fault tree was constructed using the other road users and infrastructure failure probabilities. The top-level event for the second fault-tree model was “failure of autonomous vehicle due to infrastructure components.” This model includes failure of the autonomous vehicle due to other road users, weather, construction zones or road conditions. The infrastructure-focused fault tree is illustrated in Figure 22. After allocating the failure probabilities of transportation infrastructure components it was found that the failure probability of autonomous vehicle could be 0.01571% per mile of travel.

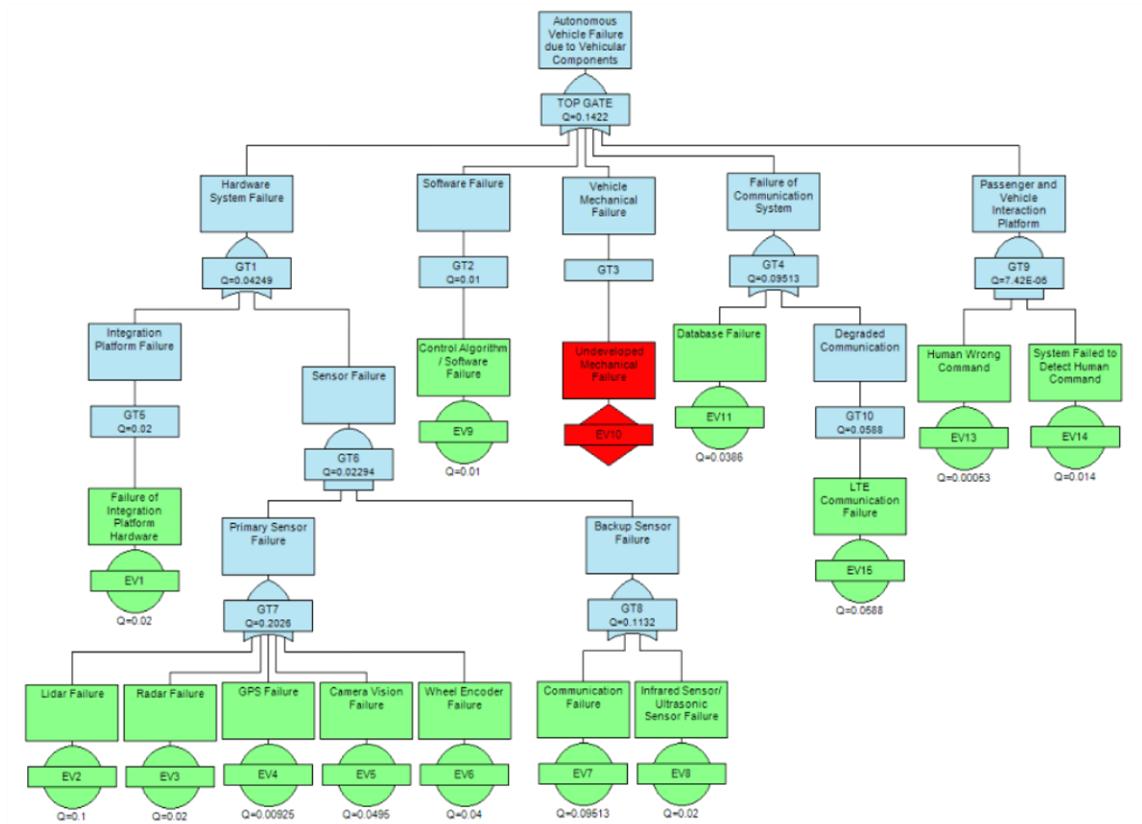


Figure 21. Fault tree analysis considering failures due to vehicular components

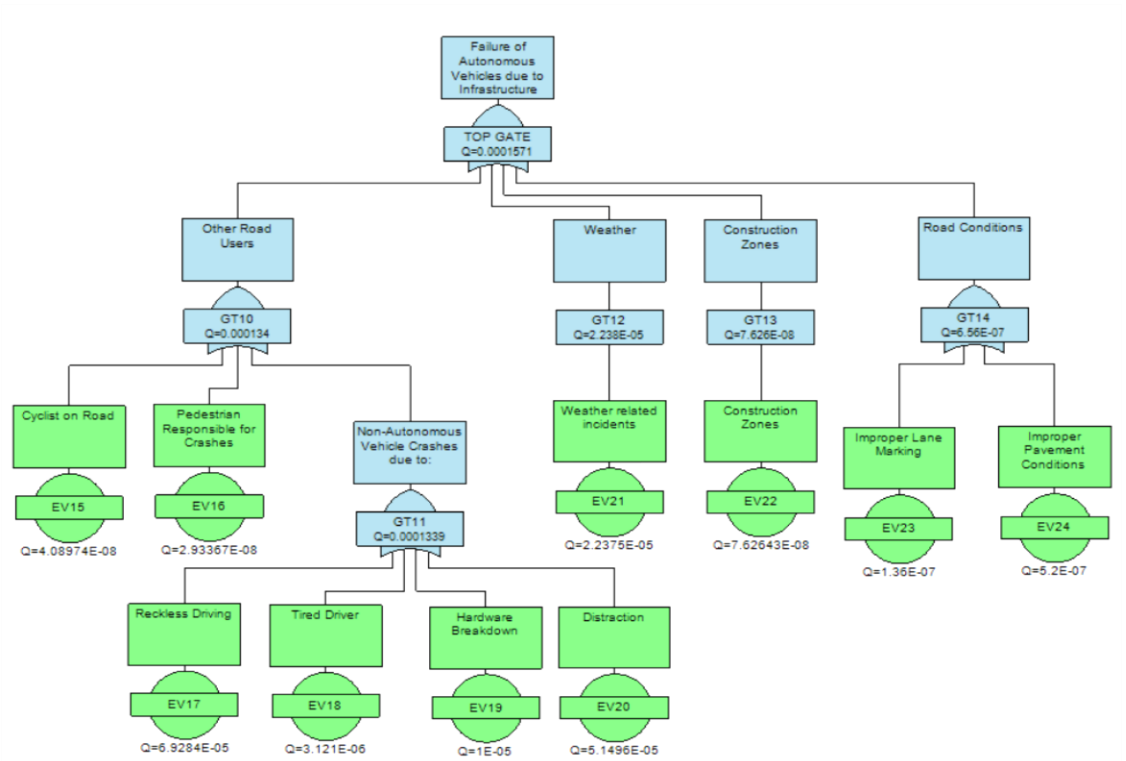


Figure 22. Failures due to transportation infrastructure components

**Combined fault tree.** The sources of all the vehicular component failures and also transportation infrastructure component failures were different. It is important to mention that few probabilities were estimated after field experiments and where others calculated probabilities based on mathematical modeling and simulation. However, combining these two fault trees, i.e., considering vehicular component failures and transportation infrastructure failures is the next step of this research. This follows the National Aeronautics and Space Administration (NASA) practice of estimating failure probabilities of basic events by applying different methods, including experimental estimation and simulation modeling (H. Dezfuli et al., 2011). Opinions of subject matter experts are also considered in probability estimations (Safie, Stutts, & Huang, 2015). The risk analysis of NASA's missions often involves the integration of various risk models,

which include failure probabilities computed by applying various methods (H. Dezfuli et al., 2011; Safie et al., 2015). Similarly, to estimate the failure probability of an autonomous vehicle travelling in a mixed traffic stream, the two fault trees developed were combined to calculate combined results of failure due to failure probabilities of autonomous vehicular components and transportation infrastructure components estimated through their respective fault-tree models (illustrated in Figure 23) as described below.

The failure probabilities of individual vehicular components collected from literature were presented early in this chapter. However, when these components become parts/subsystems of an autonomous vehicle, the car manufacturer will ensure that they remain operational throughout the life of the vehicle with periodic health monitoring and maintenance. Typically a conventional vehicle can be driven for 150,000 miles in its lifetime (Lu, 2006). Based on this information, it was assumed that the life of an autonomous vehicle is also 150,000 miles, and this assumption was used to estimate an autonomous vehicle failure probability per mile. Given that the overall probability of an autonomous vehicle failure in its lifetime is due to vehicular components the failure probability was 14.22%. The failure probability per mile can be estimated as 0.0000948% (i.e.,  $14.22\%/150,000$ ). However, the failure probability of this vehicle due to transportation infrastructure components is calculated at 0.01571% per mile, as mentioned previously. Furthermore, these two fault tree models were combined into one fault tree to estimate the overall failure probability of an autonomous vehicle due to vehicular component failures and transportation infrastructure failures in mixed traffic streams. It was assumed that the failure due to vehicular components and failure due to

infrastructure components were independent of each other and can be combined with an ‘OR’ gate to estimate the failure probability of overall autonomous vehicle system. The following equation was used to calculate the failure probability for the top-level event (i.e., failure of an autonomous vehicle) of the combined fault tree. The ‘+’ sign in the equation represents the ‘OR’ gate. As shown in the following equation, an autonomous vehicle operation could fail 158 times in 1,000,000 miles of travel due to failure of either vehicular components or infrastructure components in a mixed traffic stream. The combined fault tree is shown in Figure 23.

$$P(A) = P(VC) + P(IC) = 0.000000948 + 0.0001571 = 0.000158048 \text{ per mile of travel} \quad (13)$$

where,  $P(A)$  = Overall failure probability of autonomous vehicle system per mile of travel

$P(VC)$  = Autonomous vehicle failure due to vehicular components per mile of travel.

$P(IC)$  = Autonomous vehicle failure due to infrastructure components per mile of travel.



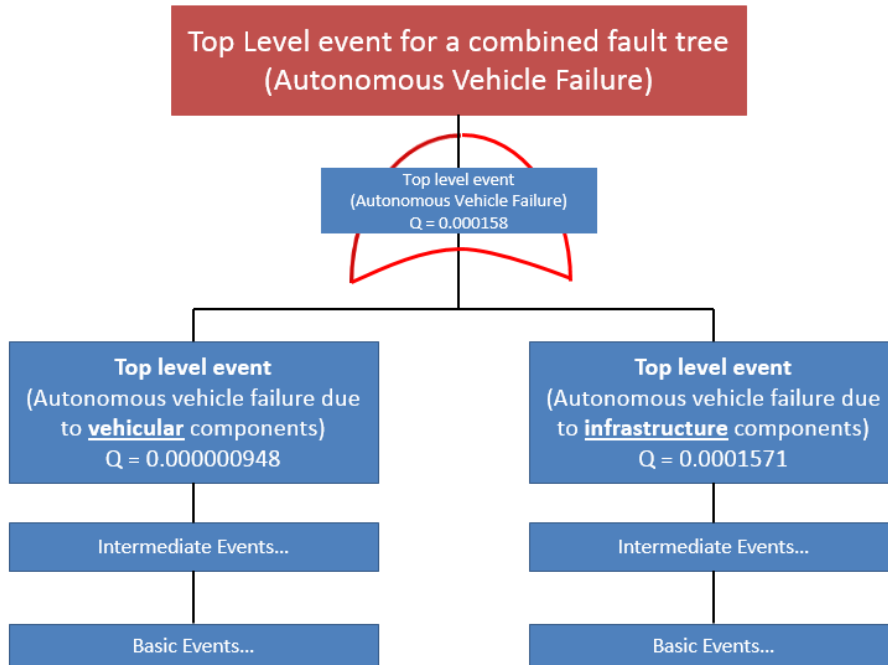


Figure 23. Failure of autonomous vehicles in mixed traffic streams using fault tree models

### Risk Hierarchization

Along with determining failure rates, a fault tree allows for cut sets to be identified within the tree which is the direct path from a basic event to the top-level event. Once all cut sets are calculated the fault tree becomes valuable. The cut set also allows engineers to determine which components to address in order to improve the performance of an autonomous vehicle. The cut sets that are particularly important are the “minimum cut set,” which exposes the basic level component because its failure will lead to a top-level failure in the shortest amount of time. This mathematical method was used to identify all combinations which are essentially the hierarchical sequence of events that can result in the failure of the main event. The logical relationships between top level and basic event are transformed using Boolean algebra, where all basic event failures are considered binary in nature, i.e., either working or failed. Notably, all component failures

were assumed to be independent, and failure rates were constant over time. Cut-sets also help decision makers to prioritize which components need to be addressed first to improve the safety performance of an autonomous vehicle. Once all cut-sets are identified, they can be ranked with associated failure probabilities.

Ten cut-sets were distinguished in the analyzed fault trees considering the failure probabilities of vehicular components and infrastructure components with the use of Isograph FaultTree+ software. These cut-sets were ranked in order of their failure probabilities. For example: hardware system failure could occur due to integration platform failure or sensor failure, while sensor failure will fail if the primary sensor and back sensor fail. Algebraic representation is given below:

$$Q = P(IP) \cup P(S) = P(IP) \cup [P(PS) \cap P(BS)] \quad (14)$$

where,  $Q$  = Hardware system cut set failure probability

$P(IP)$  = Integration platform failure probability

$P(S)$  = Sensor failure probability

$P(PS)$  = Primary sensor failure probability

$P(BS)$  = Backup sensor failure probability

Table 6 presents ranked cut-sets with their failure probabilities. It was found that the failure of the communication system could be the most vulnerable event of all the basic events with a failure probability is 9.513%. Hardware system failure, which is caused by sensitive sensor and actuator failures, was found in the second position with a failure probability of 4.249%.

Table 6

*Minimal cut-sets of autonomous vehicles components*

Rank	Cut-sets	Boolean Expression	Failure Probability
1	Communication System (GT4)	EV11+EV12	9.5130%
2	Hardware System (GT1)	EV1+ [(EV2+ EV3+ EV4+ EV5+ EV6) * (EV7+EV8)]	4.2490%
3	Software System (GT2)	EV9	1.0000%
4	Non-autonomous Vehicles Crashes (GT11)	EV17+ EV18+ EV19+ EV20	0.0134%
5	Weather (GT12)	EV21	0.0022%
6	Vehicle-passenger interaction (GT9)	(EV13*EV14)	$7.4200 \times 10^{-4}$ %
7	Road Condition (GT14)	EV23+EV24	$6.5600 \times 10^{-3}$ %
8	Construction zones (GT13)	EV22	$7.6264 \times 10^{-6}$ %
9	Cyclists (GT10)	EV15	$4.0897 \times 10^{-6}$ %
10	Pedestrians (GT10)	EV16	$2.9337 \times 10^{-6}$ %

### Evaluation of Fault Tree Model

It is required that a fault tree analysis model developed based on failure probabilities collected from different sources should be validated both qualitatively and quantitatively. The qualitative validation method considers the basic events identification and their relationship with the top-level event(s) (M. Chowdhury, Garber, & Li, 2000; Kuzminski et al., 1995). A quantitative method includes comparing the failure

probabilities estimated through a fault-tree analysis to real-world data (Tupper et al., 2014). In this research, the results from the fault tree models were compared with the real-world data available from the California DMV autonomous vehicles testing records (Delphi, 2016; Google, 2016; Mercedes-Benz, 2016; Nissan, 2016; Volkswagen, 2016). According to California DMV autonomous vehicle testing regulations, all autonomous vehicle manufactures and developers holding a permit to test must submit accident reports within 10 days of the incidents and an additional disengagement report annually (Pinto, 2012). The summary of collected crash and disengagement data from California DMV is presented in Table 7.

Table 7

*California DMV autonomous vehicles testing data*

System Failure	Description	No of Incidents	% of Incidents	Rank	References
Hardware System	Hardware discrepancy, issue with tuning, calibration, and unwanted maneuver	288	17.8439	3	(Delphi, 2016; Google, 2016; Nissan, 2016)
Software System	Software discrepancy—unable to detect vehicle or obstacles	80	4.9566	5	(Google, 2016)
Communication System	Planner data not received, drop off on received data, communication evaluation, management failure	642	39.777	1	(Mercedes-Benz, 2016; Volkswagen, 2016)

Table 7 (continued)

System Failure	Description	No of Incidents	% of Incidents	Rank	References
Non-autonomous vehicle crashes	Non-autonomous vehicle behavior at low penetration level of autonomous vehicles	68	4.2131	6	(Delphi, 2016; Google, 2016; Nissan, 2016)
Vehicle-passenger interaction	Human too uncomfortable to continue automation	487	30.1735	2	(Mercedes-Benz, 2016)
Construction zones	Signs, hand signals, lane closures, and sudden reduction of speed	31	1.9207	7	(Delphi, 2016; Google, 2016)
Road conditions	Lane marking and adverse road surface conditions	111	6.4125	4	(Delphi, 2016; Google, 2016)
Weather	Rainy, sun glare, twilight, cloudy: poor sunlight and darkness	18	1.1152	8	(Delphi, 2016; Google, 2016)

The failure probabilities of cut-sets were compared with the percentages of each crash type reported in the California DMV reports to validate the fault tree analysis findings. Thus, these crashes represent the same basic event failures that lead to cut-sets. Figure 24 compares the ranks given to each basic system failure event by the final combined fault-tree model versus the real-world data. In Figure 24, all basic failure events are ranked in a descending order of failure probability (i.e., the failure probability decreases with the increase in rank). For example, rank of 2 for hardware system failure suggests that there is a high probability of failure due to hardware failure compared to failure due to construction zones (ranked 8).

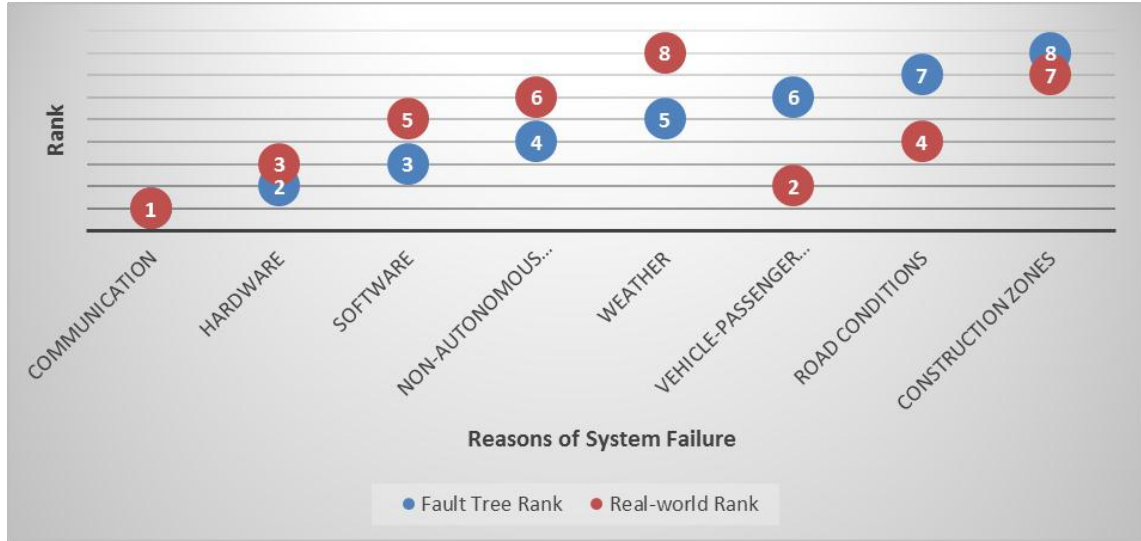


Figure 24. Comparison between the results of risk analysis and real-world incident percentages

It is found that the failure probability of communication system (ranked 1 based on the fault tree risk analysis) matches the real-world autonomous vehicle test data (also ranked 1 based on real world test data). A significant difference in the ranking of failure due to ‘vehicle-passenger interaction’ between the fault-tree analysis (ranked 6) and the real-world (ranked 2) indicates that the software system and algorithms are going through technological advancements which are captured in the fault-tree analysis but not reflected in the earlier real-world test results. Furthermore, the lower ranking (i.e., higher failure probability) using real-world data includes disengagement events reported by various car manufacturers in which the primary cause of disengagement from autonomous driving is discomfort felt by the driver (Nissan, 2016). The driver may experience discomfort and disengage from self-driving to manual driving. The possible reasoning for that could be:

- (i) The driver perceives actions taken by the autonomous mode are not safe; or
- (ii) The autonomous vehicle has failed to recognize the driver’s command.

However, with the improvement in algorithms and the increased adaptation, this discomfort may reduce, thus reducing the failure probability (Calvo-Porrall, Faña-Medín, & Nieto-Mengotti, 2017). The lower real-world rankings (i.e., higher failure probability) of weather events and non-autonomous vehicle events, in the fault-tree analysis, compared to the real-world reports suggest that autonomous vehicles have not been tested in various weather conditions and at different penetration levels.

### **Summary**

In summary, autonomous vehicles could be stopped 14.22 times over its lifetime due to the failures of vehicular components. On the other hand, the failures of infrastructure components also could lead to autonomous vehicle failure, and this failure rate was calculated as 0.01571% per mile of travel. Later, the failures of autonomous vehicles due to vehicular components and infrastructure components were combined and the overall failure rate was 0.01571% per mile of travel. The fault tree results were then validated using real-world autonomous vehicles testing data. Concluding remarks on the risk analysis of autonomous vehicles results are presented in Chapter 7. Meanwhile, the analysis and results of the online survey are represented in the Chapter 5.

## **Chapter 5**

### **Online Survey**

This chapter is divided into three sections: developing survey instruments, presenting the detailed steps needed to prepare the survey instruments, i.e. participants' list and questionnaire, and survey results. The survey results are summarized and tabulated. Finally, the survey results are analyzed using Kendall's W coefficient of concordance.

#### **Developing Survey Instruments**

The Delphi survey method was first introduced for handling the opinions of a group of experts on national security issues; however the application of this survey method has experienced different stages of development and modification (Rieger, 1986). This method can be utilized as a judgement, decision-making aid or a forecasting tool, where the subjective judgements of individuals could benefit from this method of problem solving (Gregory J. Skulmoski, 2007). The Delphi method can also guide when there is incomplete knowledge about a problem (Mbakwe, Saka, Choi, & Lee, 2016). Furthermore, the method developed for this research focuses on consensus building among the participants. Although there are variations in the survey focuses and techniques, four basic characteristics of this survey method usually remain same (Rowe & Wright, 2001); they are: i) anonymity, ii) iteration, iii) controlled feedback, and iv) statistical group responses.

A flow chart of this survey is shown in Figure 25. The experts were grouped into three panels based on their areas of expertise, since these groups have different perspectives. The three panels were 1) academic researchers' panel, 2) autonomous



vehicle industry researchers, and 3) an experts' panel from component companies, including expert researchers from automated navigation sensor companies.

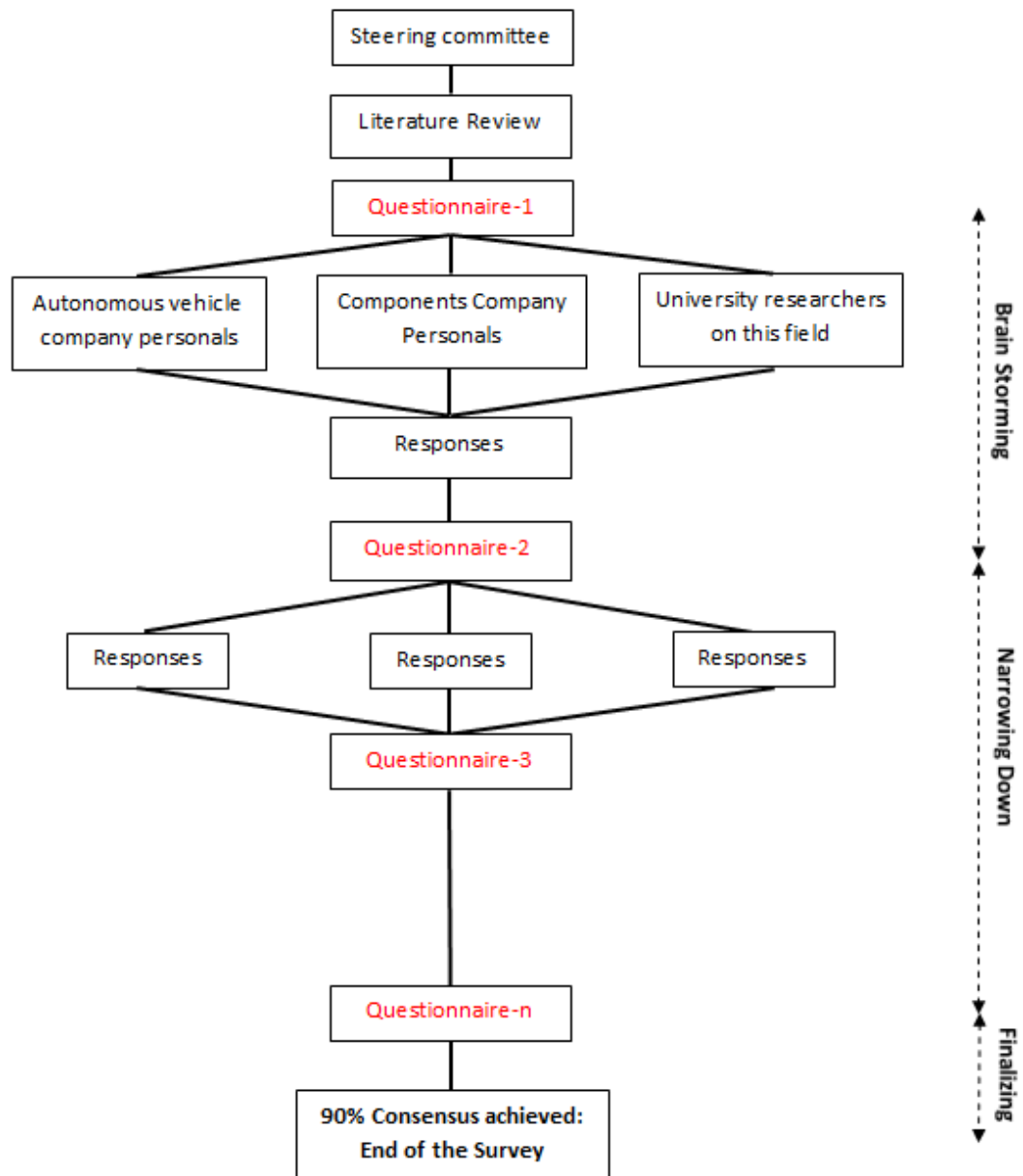


Figure 25. Autonomous vehicle Delphi survey flow

## Survey Results

A total of 140 people were invited for participation in the first round of survey distribution. However, among the invited participants only seven experts responded in this round: 50% of the responders were university researchers, 20% were researchers in industry including the manager of a development team. In the second round, about 40% of the survey participants had “more than 9 years” experience working in the autonomous vehicle research field, and another 25% had “5–9 years” of working experience.

Survey participants were asked to identify the primary sensor failure which could lead to overall autonomous vehicle failure. About 85% of the participants agreed that LIDAR and camera vision could impact the success rate of autonomous vehicle navigation, while 55% believed the GPS systems could be vulnerable to failure. The participants varied widely in their selection of failure probabilities for different vehicular components and transportation infrastructure components. For example, 60% of the participants agreed that the failure probability of LIDAR could be between 3.01 and 6.00%. For camera vision, responses from 20% based their failure probability ratios on three options: 1.01 to 3.00%, 3.01 to 6.00%, and 6.01 to 10.00%. The remaining 40% selected “greater than 10.00%.” Moreover, 50% of the responders selected the failure probability of the wheel encoder to be between 1.01 and 3.00%, where earlier it was found that the failure probability of the same wheel encoder was 4.00% based on our literature review. Even though around 60% thought communication system failure could fail the overall autonomous vehicle system, none held DSRC failure responsible. LTE communication failure was selected instead. However, participants also agreed that autonomous vehicles could be vulnerable to software and human-machine interaction

system failures. Table 8 represents the failure probabilities selected regarding vehicular component failures by the participants in the first round of the survey. Percentages of participants selected each range of failure probability shown.

Table 8

*Results of first round of survey*

Vehicular Components	Failure Probability Ranges (in questionnaire)					
	< 1.00	1.01-3.00	3.01-6.00	6.01-10.01	> 10.01	Other
LIDAR	0	0	60%	0	40%	0
Radar	0	100%	0	0	0	0
Camera	0	20%	20%	20%	40%	
GPS Device	25%	0	0	0	50%	25%
Wheel Encoder	0	50%	50%	0	0	0
Integration Platform	0	25%	25%	25%	0	25%
LTE Network	0	0	60%	20%	0	20%
Software system	0	60%	20%	0	0	20%
Database/ server	0	33%	33%	0	0	33%
Human-machine Interaction Platform	50%	25%	25%	0	0	0

Among the infrastructure components, the weather, non-autonomous drivers, cyclists and pedestrians were considered as the reasons for autonomous vehicles failure by the maximum number of participants (about 70%). However, the participants provided a wide range of failure probabilities for these infrastructure components.

### **Analysis of Survey Results**

Researchers considered consensus measurement as a viable component of data analysis and interpretation in research, which measure the level of agreement achieved

among the expert panel. However, consensus measurement also utilized a stopping criterion of iteration, where group stability and individual stability were used as the necessary criterion in many studies. Even though, many researchers suggested that consensus measurement does not match with the original idea of the Delphi survey method, the measurement parameter could be deployed in achieving agreement over qualitative outcomes. However, to draw conclusions for quantitative outcomes, inferential statistics could be utilized based on data and the normal frequency distribution of dataset. Depending on whether the dataset followed a normal distribution, parametric and nonparametric tests have been used in Delphi studies. Many methods can be utilized to analyze the Delphi survey results and to calculate the level of consensus. For example, the chi square test, McNemar's change test, the Wilcoxon matched-pairs signed-rank test, Spearman's rank-order correlation coefficient, Kendall's W coefficient of concordance and F tests. In this research, Kendall's W coefficient of concordance was used to measure the level of consensus between two consecutive rounds of Delphi surveys (Cafiso, Di Graziano, & Pappalardo, 2013). Table 9 shows the interpretation of Kendall's W adopted in this study.

Table 9

*Interpretation of Kendall's W*

Kendall's W	Interpretation
$W \leq 0.3$	Weak agreement
$0.3 < W \leq 0.5$	Moderate agreement
$0.5 < W \leq 0.7$	Good agreement
$W > 0.7$	Strong agreement

The Kendall's W coefficient of concordance was utilized to calculate the level of consensus, and we decided to continue the iteration till strong agreement was achieved (Kendall's W equals to 0.7 or higher). For instance, 3 out of 5 participants selected 3.01 to 6.00% as the failure probability of LIDAR, and others selected greater than 10.00%.

Null Hypothesis: There is no agreement among the participants upon the failure probability of LIDAR.

Alternative Hypothesis: The participants agreed upon the failure probability of Lidar.

For this hypothesis, Kendall's W was 0.8 for the question concerning LIDAR failure probability. This suggests "strong agreement" among the participants. Also, the one-tailed p-value was 0.00302, which indicates no agreement among the participants to reject the null hypothesis. Detailed calculation is provided in Appendix D.

Similarly, Kendall's W was calculated for the failure probability of camera vision. The value of W was equal to 0.2 which represents "weak agreement" among the participants. With a one-tailed p-value of 0.41, it is very likely that no agreement was reached among the experts.

## **Summary**

In summary, the online survey was conducted to include the experts' opinions in risk analysis of autonomous vehicles. Even though, 140 experts were identified and invited for their participation in the survey, only seven people responded in time. Due to low response rate, the survey results could not be utilized to draw any strong inference. Further remarks are presented in Chapter 7, and autonomous vehicle\ microsimulation results are provided in next chapter.

## **Chapter 6**

### **Autonomous Vehicle Simulation Results**

After developing the autonomous vehicle navigation algorithms, traffic simulation models were being simulated to evaluate the performance of these advanced vehicles on our roads in mixed traffic environment. This chapter focuses on analyzing the simulation results and estimating the overall safety accomplishments over the replacement of human drivers with autonomous vehicle on roads. The chapter is divided into two sections. In the first section, the results of crash frequency estimation are described. Later, the impacts of autonomous vehicle crashes on the performance of transportation infrastructure are presented.

#### **Crash Frequency Estimation**

In the Vissim traffic simulation software, Interstate-476 (I-476) was sketched and calibrated with the real-world traffic volumes where the autonomous vehicle penetration level is zero. This model was considered as a base model and compared with the models where different market penetration levels of autonomous vehicles were coded. In this research, the autonomous vehicles market penetration of 10%, 25%, 50% and 90% were modeled as mentioned in Chapter three. The automated platooning was programmed as the driving feature of autonomous vehicles. This feature was embedded in simulation using the dynamic link library (DLL) file, developed earlier and first mentioned in Chapter 3. The Vissim model exported the vehicle information, i.e., current speed, acceleration, and the speed difference between a leading and corresponding vehicle, to a DLL file. Then, the DLL file evaluated the information imported from the Vissim models and calculated the next maneuver of autonomous vehicles using the defined cases in

Chapter three. Finally, the DLL file forwarded the corresponding values to Vissim to execute the next simulation second in a microsimulation environment.

Researchers use different transportation parameters, i.e., travel time, queue length, density, and delay as road network performance measures in transportation projects. However, travel time data is the most preferred one among them, as this parameter can be utilized in transportation planning, operations, management, maintenance, and evaluations. Also, in this research, travel time was estimated and evaluated to compare the performances of the overall transportation infrastructure after deploying autonomous vehicles on roads and highways. Five travel time measurement segments were modeled in Vissim to estimate average travel time over a certain period of time, i.e., 900 seconds. The demographic location of these five travel time measurement segments are provided in Figure 26 and their lengths (in ft) are in Table 10.

Table 10

*Description of modeled travel time measurement segments*

Segment Number	From Exit	To Exit	Distance (ft)
1	63: I-476 South Mainline between the Exit 9 off and on ramps	80: I-476 South Mainline between the Exit 5 off and on ramp	19256.77918
2	80: I-476 South Mainline between the Exit 5 off and on ramp	94: I-476 South Mainline between the Exit 3 off and on ramp	9183.832427
3	94: I-476 South Mainline between the Exit 3 off and on ramp	110: I-476 South Mainline Before Hwy I95 (3 lanes)	15326.8873
4	12: I-476 North before the Exit 1 on ramp	30: I-476 North Mainline between the Exit 3 off and on ramps	15304.756

Table 10 (continued)

Segment Number	From Exit	To Exit	Distance (ft)
5	30: I-476 North Mainline between the Exit 3 off and on ramps	42: I476 North Mainline between the Exit 5 off and on ramps	9173.011851

The traffic volumes assignment in Vissim follows a stochastic distribution (PTVGroup, 2015). This distribution is set up so that a specific time dependent vehicle can enter a link in a distributed manner. The time gap between two successive vehicle entrances depends on the assigned hourly traffic volume. However, a random number generator is used to estimate the time gap values from the software stochastic distribution. In Vissim, a parameter called “random seed” actually initializes the random number generation (PTVGroup, 2015). It is important to increment this random seed number to capture the variability of traffic patterns. In this research, the initial value of random seed was assigned as 5, and then this value is incremented by 5 in each simulation run. Based on the calculation of the simulation run number, presented in Chapter 3, a total of 11 simulation runs were executed for each autonomous vehicle market penetration level.

Each simulation model ran for a period of 4800 simulation seconds, where the initial 600 seconds and last 600 seconds were utilized as “warm up” and “cooling off” time. These warm up times ensured enough time to fill up the road network and the cooling off times provided time to dissolve the queue formed in the simulation period. The simulation seconds in between warm up and cooling off times were divided into four segments of 15 minute-time intervals.



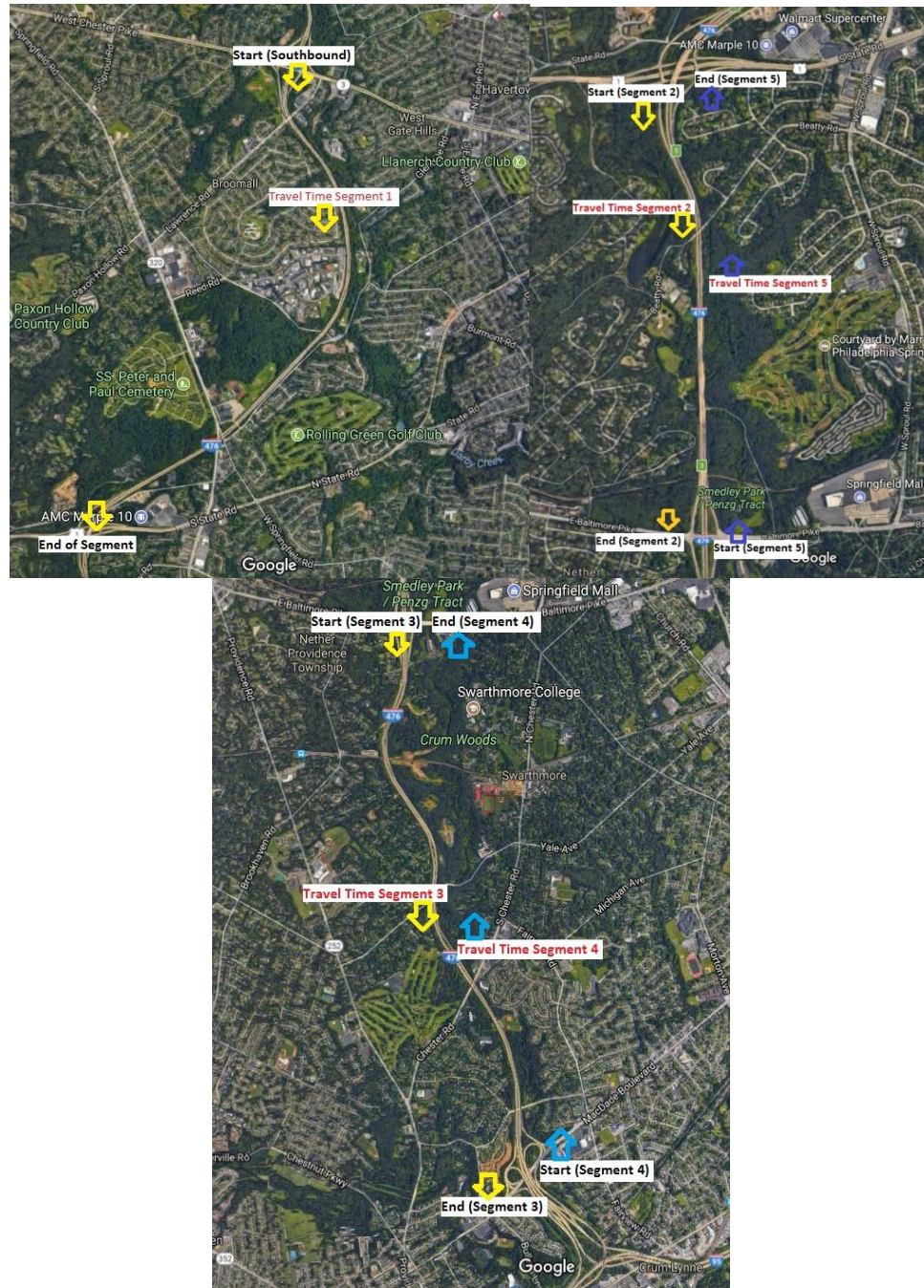


Figure 26. Simulated travel time measurement segments (Source: Google Map)—not to scale

The travel time for each 15-minute interval was recorded for each of the total 11 simulation runs for the base model, where the autonomous vehicle percentage was zero to total vehicles. The same step was followed for the rest of the simulation models, where

autonomous vehicle percentages varied in between 10 and 90. Each of the five travel time measurement segments were analyzed and their results were stored accordingly.

The raw travel time values were recorded for travel time measurement segments and are in Appendix E. The travel times cover an interval of 600 to 4200 simulation seconds with different autonomous vehicle market penetration levels. Also variations in travel times due to different random seed number were tabulated. However, it is difficult to draw patterns of travel time variations over the random seed numbers, because a random seed number represents different portions under the distribution curve. Later, the travel time values for a single penetration level are averaged arithmetically over 11 runs.

The average travel time over simulation runs with different random seed number were then compared. It was found that travel time increased from the time interval of 600 to 1500 seconds to 3300–4200 seconds, due to increase of queue length. However, a certain drop of travel times occurred in time intervals of 2400–3300. The average travel time for different penetration level is compared in Figure 27.

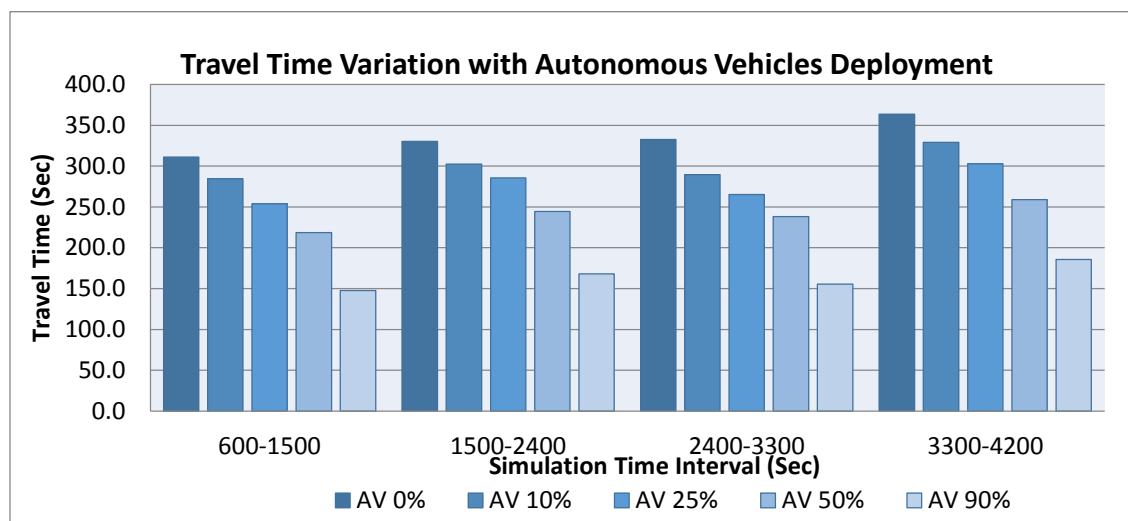


Figure 27. Average travel time over different random seed numbers

It is expected that deployment of autonomous vehicles will reduce traffic congestion and increase the roadway capacity, thereby reducing the overall travel time to reach from origin to destination. Figure 27 shows that travel time was reduced after deployment of autonomous vehicles on roadways. However, the reduction of travel time from the base model was calculated for four autonomous vehicle market penetration levels, i.e., 10%, 25%, 50% and 90%. It was found that travel time values were reduced on an average of 8 to 9% after implementing 10% autonomous vehicles on roads. The reduction of travel times increased with the increase of autonomous vehicle penetration levels. Figure 28 shows the percentage of travel time reductions for travel time Segment 1. This figure shows that an autonomous vehicle can deduce the travel time by about 51% with a market penetration level of 90%.

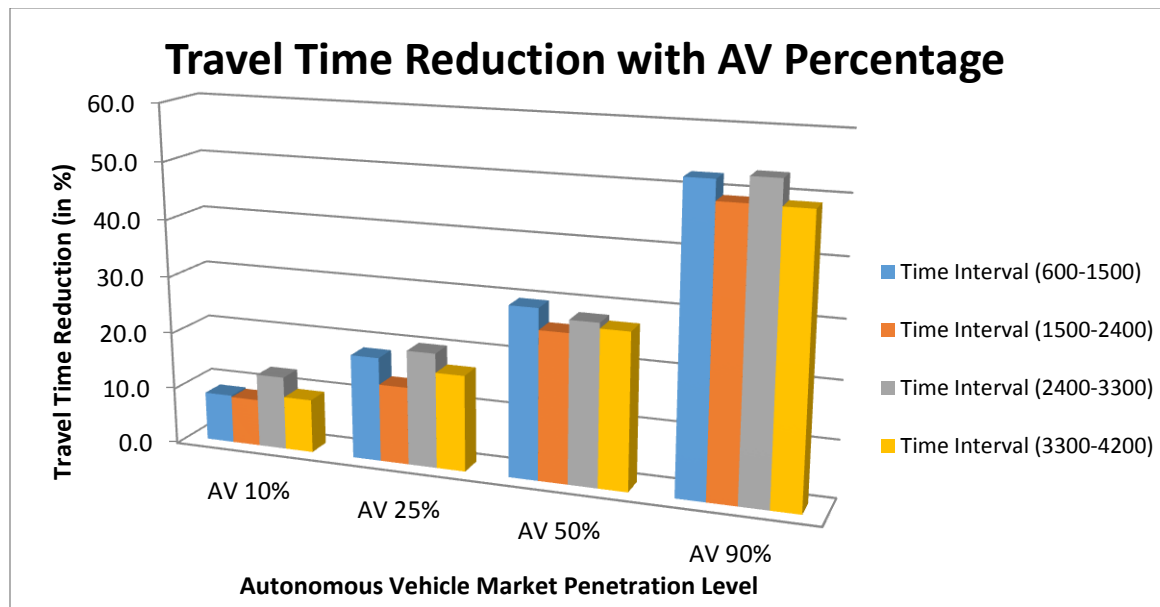


Figure 28. Travel time reduction percentages over autonomous vehicle market shares for travel time measurement in segment 1

Similar steps were followed for the rest of the four travel time segments. For travel time Segment 2, the travel time reductions varied from range 4% to 51%. With 10% autonomous vehicles, travel times were reduced by about 4% and those values were reduced by about 10 and 30% with 25 and 50% autonomous vehicles. The reductions in travel time for Segment 2 are plotted in Figure 29.

Figure 30 shows that the travel time reductions for travel time Segment 3 after autonomous vehicle deployment followed similar trends as noted for previous segments. A 14% travel time deduction went into effect after implementing autonomous vehicles as 10 percent of total vehicles. These travel time reductions increased over the increments of the autonomous vehicle market penetration levels.

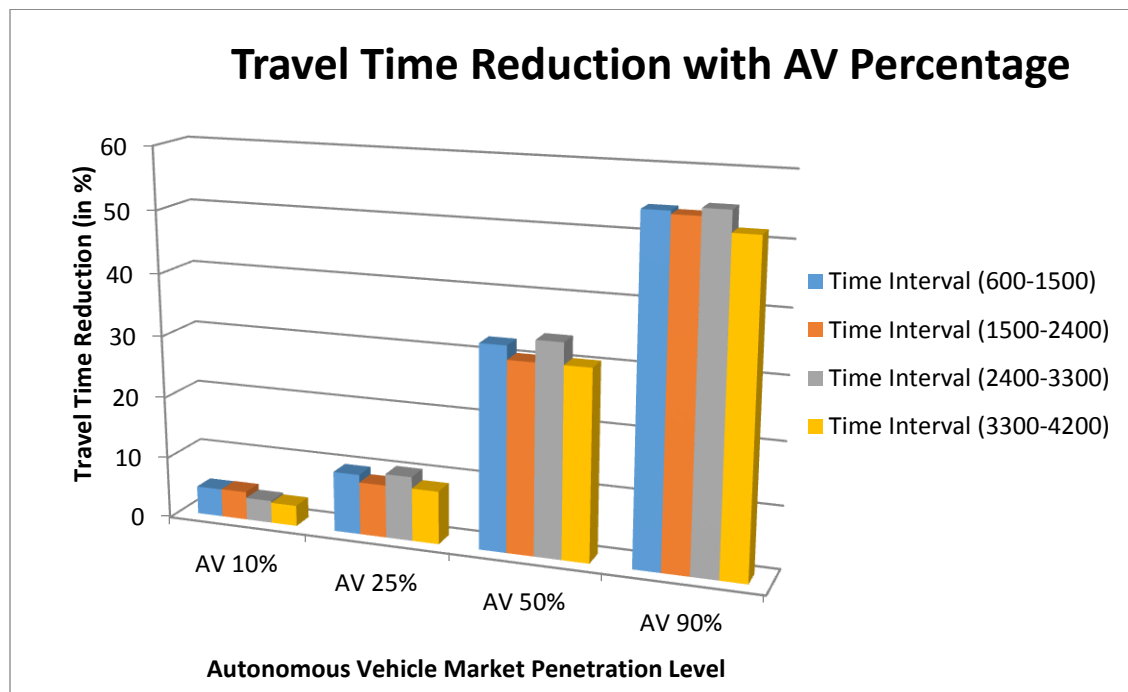


Figure 29. Travel time reduction percentages over autonomous vehicle market shares for travel time measurement in segment 2

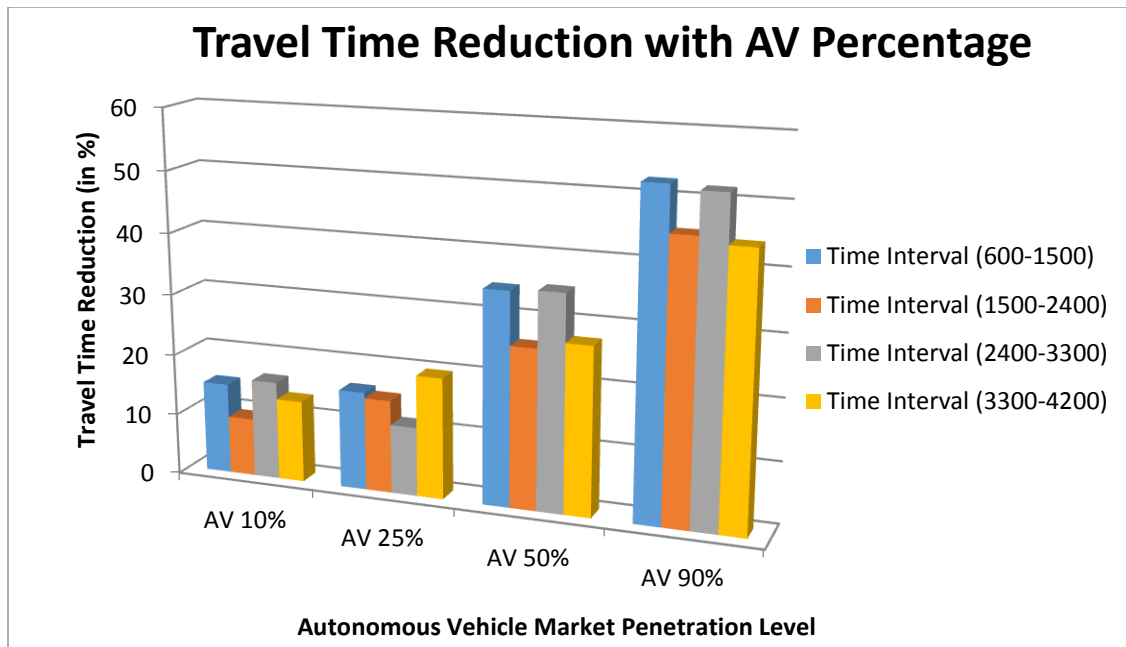


Figure 30. Travel time reduction percentages over autonomous vehicle market shares for travel time measurement in segment 3

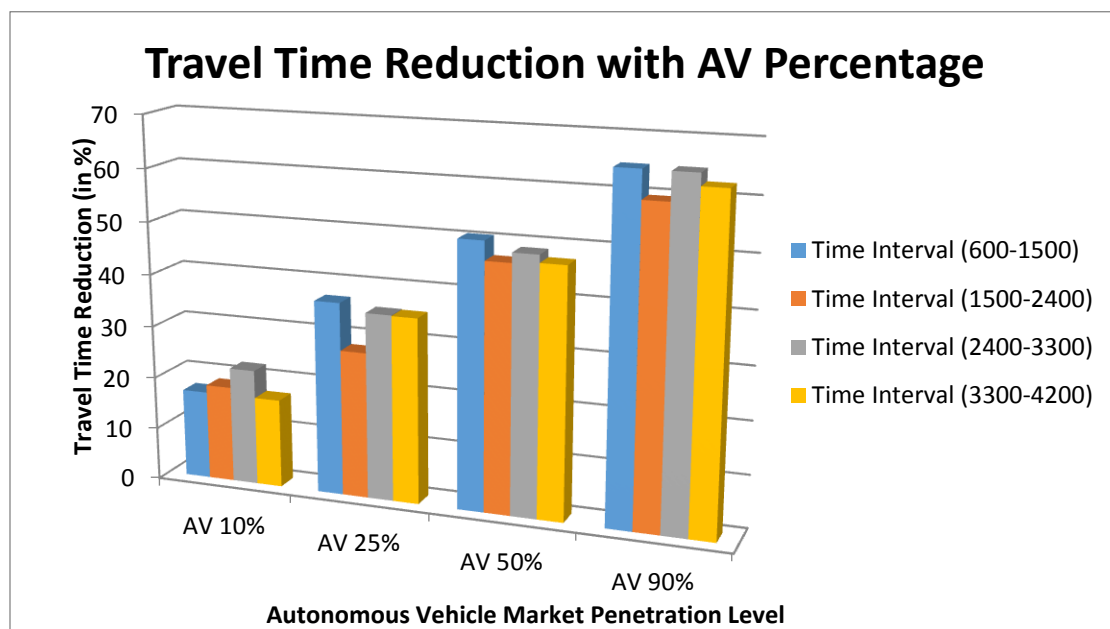


Figure 31. Travel time reduction percentages over autonomous vehicle market shares for travel time measurement in segment 4

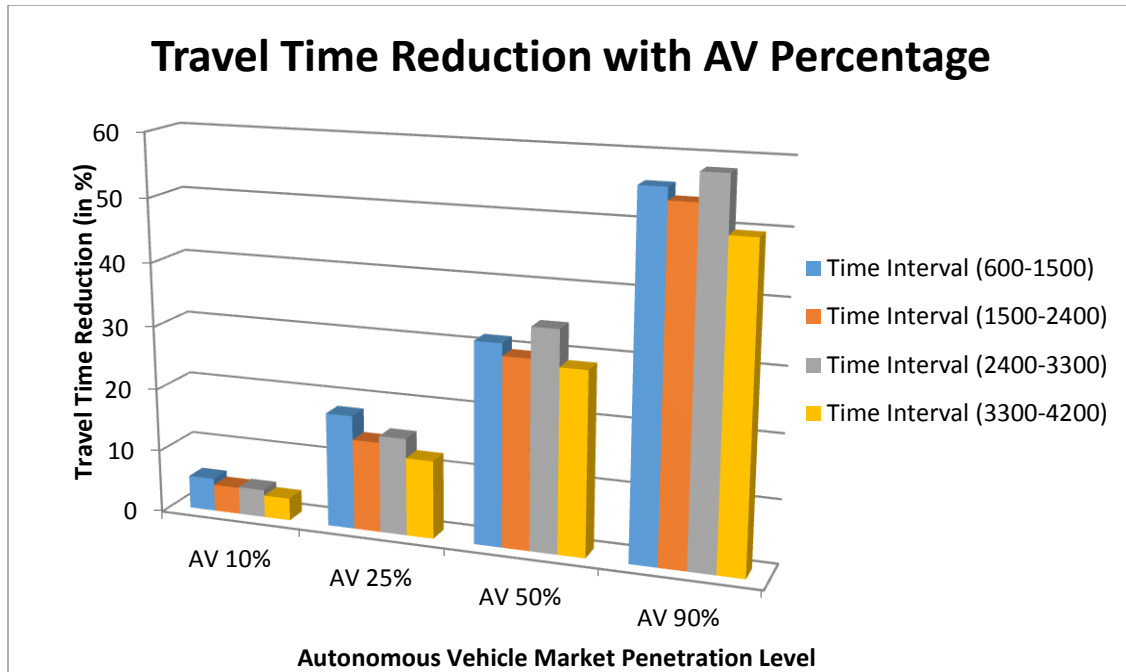


Figure 32. Travel time reduction percentages over autonomous vehicle market shares for travel time measurement in segment 5

The travel time values were lowered for travel time Segments 4 and 5 by increment of the autonomous vehicles market shares, shown in Figure 31 and 32 respectively. For travel time measurement Segment 4, the overall travel time was trimmed by about 62% with an autonomous vehicle penetration level of 90%, and this value was around 54% for travel time Segment 5.

After the performance evaluation, the trajectory files developed during Vissim simulation runs were imported into SSAM software. Five models were developed in this software for five autonomous vehicle penetration levels. In this research, three types of conflicts were considered for safety evaluation, crossing conflicts, lane change conflicts and rear end conflicts, as mentioned in Chapter 3. However, the default values of TTC and PET were utilized first. The 11 trajectory files were imported in each SSAM model, where each model represents a single autonomous vehicle penetration level. In SSAM,

each trajectory file was evaluated separately and recorded. The conflicts analysis results are presented in Appendix F, where TTC = 1.5 seconds and PET = 4.0 seconds.

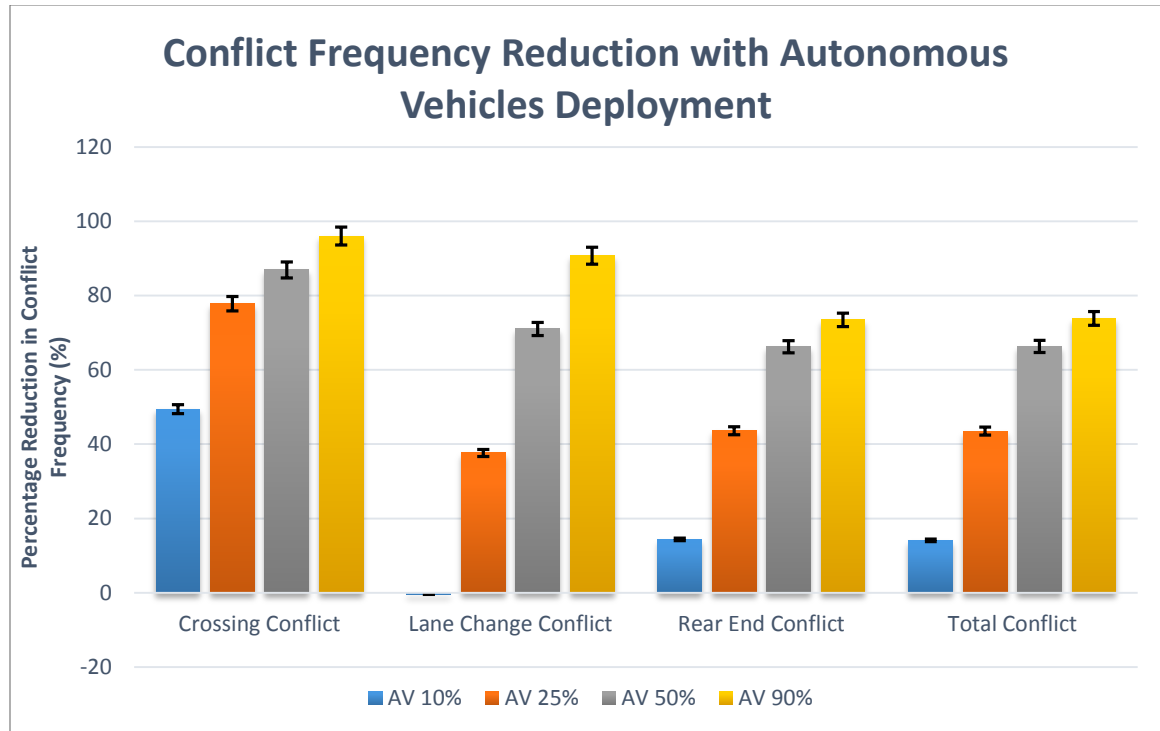
After analyzing trajectory files in SSAM, the results were validated using student's t-test. To perform this statistical test, two autonomous vehicle penetration levels were considered, they were 0% and 10%. The null hypothesis was the mean of total conflicts calculated from 11 simulation runs for 0% autonomous vehicle penetration level was equal to the mean of total conflicts calculated for 10% autonomous vehicle penetration level. With 95% confidence level, it was found that the mean of total conflicts for 10% autonomous vehicle penetration level was estimated to be lessened than the same value for 0% autonomous vehicle penetration level (with t-statistic = 5.8045 and two-tail p-value =  $1.115 \times 10^{-5}$ ). The means of total conflicts were 143,677 and 123,455 respectively for autonomous vehicle penetration level 0% and 10%.

It was found that total numbers of conflicts were decreased with the increase of autonomous vehicle market penetration levels. However, the number of lane change conflicts increased by 157 conflicts after moving to the 10% autonomous vehicles scenario from the 0% autonomous vehicles included. The possible reasoning is autonomous vehicles were engaged more on lane changing behaviors as the chances of platooning were low due to low autonomous vehicle penetration. The comparison of the estimated conflicts among different autonomous vehicle penetration levels with 95% confidence is shown in Figure 33.

The frequency of crossing conflicts reduced 49% after implementing 10% autonomous vehicles on the roadways. This reduction value increased to 96% after increasing the autonomous vehicle percentage to 90%. The frequency of lane change



conflicts increased by 0.5% initially, after implementing 10% autonomous vehicles into the total vehicle traffic mix. However, the lane change conflicts started to reduce after 25% autonomous vehicle penetration, and reduced by approximately 90% after deploying autonomous vehicles as 90% of the total vehicle traffic mix. Moreover, rear-end conflicts were reduced by 14% to 73% depending on the percentage levels of increase in the autonomous vehicle population being monitored in Chapter 3.



*Figure 33.* Conflict reduction frequency with the increase of autonomous vehicle population in mainstream traffic mix

Later, the variation of conflict frequencies with different TTC and PET values were evaluated. In this research, three values of TTC, i.e. 0.9, 1.2 and 1.5 seconds, and two values of PET, i.e., 3.0 and 4.0 seconds were utilized to generate the trend of conflict



reduction with different percentages of autonomous vehicles, i.e. 0%, 10%, 25%, 50% and 90%. The variations in conflict frequency reductions with different TTCs and PETs are presented in Tables 11 and 12.

In Table 11, conflict frequency was reduced with the decrease of TTC and retention of the same PET. When  $TTC = 0.9$  seconds, a more limited conflicts region was evaluated than when  $TTC = 1.5$  seconds, so the number of conflicts was less for  $TTC = 0.9$  seconds than for  $TTC = 1.5$  seconds. However, these numbers of conflicts were reduced by the increase of autonomous vehicle penetration levels into the mainstream traffic mix. The conflict frequency was reduced by 61% with 90% autonomous vehicles when  $TTC$  was 0.9 seconds, and this value was 68% and 73% when  $TTC = 1.2$  and 1.5 seconds, respectively.

Table 11

*Variations in conflict frequency reductions when  $TTC$  (= 0.9, 1.2 and 1.5) and when  $PET$  (= 4.0)*

AV Penetration	TTC = 0.9 & PET = 4.0		TTC = 1.2 & PET = 4.0		TTC = 1.5 & PET = 4.0	
	Total Conflicts	% Reduction	Total Conflicts	% Reduction	Total Conflicts	% Reduction
AV 0%	958362	--	1219196	--	1580449	--
AV 10%	519831	45.76	843649	30.80	1358008	14.07
AV 25%	465888	51.39	632249	48.14	892703	43.52
AV 50%	432922	54.83	469520	61.49	531988	66.34
AV 90%	366993	61.71	385112	68.41	413636	73.83

Table 12 presents similar conflicts analysis with different PET values, which was 3.0 seconds. Notably, the number of total conflicts when  $PET = 3.0$  seconds were similar as the total conflicts when  $PET = 4.0$ , but this was only when autonomous vehicle

penetration level was 0%. However, for other autonomous vehicle penetration levels, the numbers of total conflicts when PET = 3.0 seconds varied significantly and were actually lower than similar values when PET = 4.0 seconds. With 90% autonomous vehicles, the total conflicts were reduced by 61, 69 and 74% when TTC values were 0.9, 1.2 and 1.5 seconds respectively, while PET values remained same as when PET = 3.0 seconds.

Table 12

*Variations in conflict frequency reductions when TTC (= 0.9, 1.2 and 1.5) and when PET (= 3.0)*

AV Penetration	TTC = 0.9 & PET = 3.0		TTC = 1.2 & PET = 3.0		TTC = 1.5 & PET = 3.0	
	Total Conflicts	% Reduction	Total Conflicts	% Reduction	Total Conflicts	% Reduction
AV 0%	958362	--	1219196	--	1580449	--
AV 10%	475523	50.38	654373	46.33	982021	37.86
AV 25%	444343	53.64	539091	55.78	707800	55.22
AV 50%	428933	55.24	451511	62.97	493747	68.76
AV 90%	365144	61.90	376723	69.10	396734	74.90

Researchers identified that the number of conflicts calculated using SSAM can be significantly correlated with actual crash data (Archer, 2005; Dijkstra et al., 2010; Gettman & Head, 2003). However, Vogt utilized coefficient of determination ( $R^2$ ) to measure this correlation between SSAM predicted conflict results and actual crash data, and found that it varied within a range of 0.31 and 0.51 depending on road segment type, with an average of  $R^2 = 0.41$  for all road types, i.e. urban and rural (Vogt, 1999). Additionally it was found that a mean absolute percentage error (MAPE) value of 18% in prediction performance of SSAM models (Huang et al., 2013). It is also important to mention that even for same road segments SSAM predicted different values of conflicts

when trajectory files are generated from different traffic simulation software, i.e. Vissim, Aimsun, Paramics and Texas. For instance, a study found that SSAM estimated 10 times more conflicts after analyzing imported files from Texas than Vissim (Gettman & Head, 2008). Though, SSAM prediction models demonstrate a success in analysis of proposed traffic facilities and comparison between two alternatives, the results of these models are not definitive, more likely qualitative. It is recommended that the SSAM should be utilized to rank the proposed surrogate measures, rather than estimating number of crashes (Gettman & Head, 2008).

### **Integration of Fault Tree and Simulation Modeling**

The second phase of simulation analysis was measuring the impact of autonomous vehicle failures in mixed traffic streams. The similar DLL file was utilized here to simulate autonomous vehicle platooning. In this phase, the autonomous vehicle penetration level was considered 10% as the fault tree risk analysis models were developed assuming an autonomous vehicle penetration level of 10%. However, an additional visual basic code was used to monitor and control the maneuvers of autonomous vehicles. In fault tree models, it was determined that autonomous vehicles could fail to navigate autonomously 158 times in one million miles. Based on this information it could be reported that autonomous vehicle can drive an average 6,329 miles, i.e.,  $1,000,000/158$ , before a failure occurs. As mentioned in chapter three, an autonomous vehicle was randomly selected to fail, when total distances covered by all vehicles exceeded 6,329 miles. It was not possible to simulate one vehicle to cover the entire length of 6329 over a certain period within the limited the length of roadways in

the selected study region. To resolve this issue, the distances covered by all vehicles were selected to control autonomous vehicle failure scenarios.

The simulation was run between 600 to 4200 simulation seconds; however, the time interval used was 100 seconds in this phase to capture a more accurate variation in travel time. The random seed value was assigned 1, and that value remained the same in both crash and non-crash scenarios. However, within the time frame and the limited region of roadways there were two simulated crashes modeled on the monitored roadways, when the total distances covered by all autonomous vehicles were 6,329 and 12,865 respectively. It was assumed that the crashed vehicle would remain at velocity = 0 mph on the incident location for 180 seconds before the emergency response team appeared. After this response period the crashed vehicle would be removed from the traffic network using a Visual Basic code.

The travel time results for travel time measurement Segment 1 collected from Vissim are presented in Figure 34. The travel time values were exactly equal for two simulation runs (since they both had the same random seed number), until the failure of the autonomous vehicle occurred. The first autonomous vehicle incident occurred on another travel time segment path (not on the Segment 1 travel time), so that incident did not impact the Segment 1 performance. However, the second failure happened in the Segment 1 time interval of 3700-3800 and that impacted travel time results. The travel times between the failure and non-failure scenarios varied by 0.24%, and this was significantly lower. However, large scale modeling with multiple crash scenarios led to some conclusions.

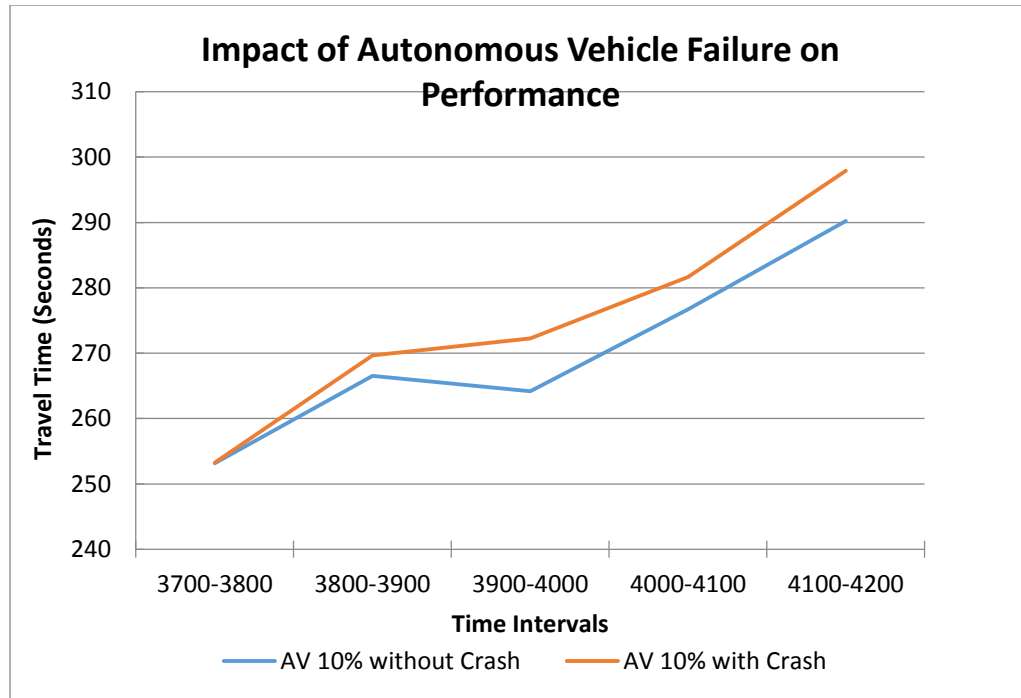


Figure 34. Variation of travel time between failure and non-failure scenarios

## Summary

In summary, the autonomous vehicles' microsimulation results were presented in this chapter. It also covered the safety and operational benefits of deploying these vehicles on our roads and highways. To introduce randomness in simulation, random seed values were varied within a wide range of 5 to 55. Thus, the travel time from origin to destination could be reduced by 50% after deploying 90% autonomous vehicles of the total vehicles available for this study. Furthermore, traffic crashes could be reduced by replacing human drivers with autonomous vehicles. With a 90% autonomous vehicle penetration level, 73% of all conflicts were eliminated there by saving human lives and avoiding injuries and property damage. Remarks on the simulation results are presented in Chapter 7: Conclusions and recommendations.

## **Chapter 7**

### **Conclusions and Recommendations**

The first step of this thesis was to identify the potential sources of risks associated with the failure of autonomous vehicle navigation. The failure of any single component that could lead to the failure of the overall autonomous system was considered and evaluated. Then, the fault tree-based risk analysis method was utilized to analyze the performance of the autonomous vehicle system. The reliability of each autonomous vehicle component was determined through the comprehensive literature review. The failure probabilities of vehicular components were plugged into the developed fault tree structure and the analysis was run in the software to identify the most critical components. These component failures could lead to overall system crashes in the shortest possible time. Autonomous vehicle navigation could be stopped due to the failures of either vehicular components or transportation infrastructure components. The combined failure probability was determined to be 0.0158048% per mile of travel. Furthermore, the autonomous vehicle lifetime status value was projected to be capable of 158 failure incidents in 1,000,000 miles of travel due to failure of either vehicular components, or infrastructure components in a mixed traffic stream. These results could be used to develop risk minimization strategies to eliminate or reduce system failures and finally ensure safety to the passengers of autonomous vehicles. Furthermore, the results of fault-based risk analysis were quantitatively validated with the real-world data of autonomous vehicle testing, collected from the California DMV webserver.

However, reckless human drivers were found to be one of most critical factors affecting autonomous vehicle navigation. They are the dominant concern for autonomous

vehicles in a mixed traffic stream. Yet, at the initial stage of autonomous vehicle deployment, these advanced vehicles need to share the roads and highways with non-autonomous drivers. Based on the results of this research, the policymakers should develop certain rules and regulations to facilitate the sharing of roads and highways. Researchers recommend separate lanes for autonomous and non-autonomous drivers on multilane sections of roadways as one potential solution (Chen, He, Yin, & Du, 2017), (Turnbull, 2015). Other researchers claimed that installation of black boxes in autonomous vehicles to record the sensors data and surrounding information, could be useful for crash investigations after collisions between autonomous vehicles and conventional vehicles driven by non-autonomous drivers (Nothdurft et al., 2011). However, further research is needed to confirm the advantages and hence, the applicability of these solutions.

The second step was an online survey seeking further information of the vehicular components failure probabilities from the subject matter experts. The Delphi survey method was utilized to prepare the survey framework. The benefit of this survey method was to develop multi-round anonymous interactive participation through questionnaires. However, only seven experts responded among the 140 experts invited to participate in the online survey. The survey results showed that experts agreed “strongly” on the question asking the failure probability of LIDAR, whereas “weak agreement” was found in the case of a camera failure probability value. However, due to the small participation pool, the survey results are not recommended to represent the majority of expert’ opinions nor to draw a strong inference due to the limited number of responses.

Traffic microsimulation was carried out in the third step of this thesis. The algorithms were developed and then utilized to model autonomous navigation in a microscopic traffic simulation environment. A segment of interstate highway in Pennsylvania was modeled as the study region. Then, the traffic model was calibrated and validated to represent the real-world traffic scenarios. The gradual increase of autonomous vehicle market penetration level was drawn by using different percentages of autonomous vehicle among all transportation modes, i.e. 0%, 10%, 25%, 50% and 90%.

Five travel time segments of different lengths and directions were designed and evaluated with different autonomous vehicle penetration levels. To generate randomness in simulation results the random seed number was varied within a range of 5 to 55. After executing the simulation with different autonomous vehicle-penetration scenarios, the travel times for each 15-minute interval were recorded. After analyzing the travel time data, it was found that autonomous vehicle can reduce travel time by 51 to 64% with a 90% market penetration level. However, the trajectory files generated by traffic simulation were exported to investigate the safety of autonomous vehicles and estimate the conflict frequencies. It was found that about 73% of total conflicts which could result in a traffic crash could be avoided by replacing 90% of human drivers with autonomous vehicles. Moreover, it was found that conflict frequencies fluctuated with the change in TTC and PET values. Finally, a comparison between a failure and non-failure scenario of an autonomous vehicle was drawn to integrate fault tree analysis results in simulation.



## **Recommendations**

- In this thesis it was not possible to conduct statistical validation due to limited availability of autonomous vehicle testing data. Further research is recommended for comparing the fault tree-based risk analysis results with real-world risk analyses.
- In this research, all the vehicular components were assumed to represent an independent and individual component. The interdependency among the vehicular components was not considered. However, it is recommended that the interdependency among these components should be investigated before integrating into another fault tree analysis. Also, the developed fault tree should be revised based on the interdependency analysis.
- The failure probabilities of vehicular components were assumed to be constant over the lifetime of these components. However, the lifetime performance of these components could vary. Variation in the performance of sensors over time (i.e., time dependency on reliability) should be considered. In future research, the failure probabilities of these components should be revised based on either experts' opinions or further experimental testing.
- The final fault tree was developed by combining the developed fault tree based on vehicular component failures and the developed fault tree based on transportation infrastructure components failure. It was assumed that these two fault trees were independent. However, these fault trees could overlap depending on the nature of the critical components' failure. The interdependency of these two trees should be considered in future studies.
- A traffic simulation model was calibrated using real-world data in this research, and it was assumed that this calibration would be valid after deployment of autonomous

vehicles. In the future, traffic models should be calibrated based on road-tested autonomous vehicle trip information.

- In traffic simulation, travel demands and choice of modes were not considered; however, with the deployment of autonomous vehicles, these values should be updated and considered in traffic microsimulation.

- In the future, advanced simulators will be utilized to further analyze the safety improvement of autonomous vehicles over human drivers. Two of the currently available simulator packages are: CarSim (CarSim, 2017), and Webots (Webots, 2017). These advance simulators could provide more accurate results than the results presented in this research from the integration between traffic microsimulation and the SSAM tool. United States Patent and Trademark Office (USPTO)-registered, professional autonomous vehicle simulation packages allow coding and analyses of model vehicle dynamics as well as traffic crash scenarios, which does a better job of simulating realistic behavior.

## References

- Abraham, H., Reimer, B., Seppelt, B., Fitzgerald, C., Mehler, B., & Coughlin, J. F. (2017). Consumer Interest in Automation: Preliminary Observations Exploring a Year's Change. *AgeLab*.
- Allan F. Williams, & O'Neill, B. (1974). On the road driving records of licensed race drivers. *Accident Analysis Prevention*, 72(3-4), 260-272. doi: <http://www.sciencedirect.com/science/article/pii/0001457574900049>
- Alonso, I. P., Llorca, D. F., Sotelo, M. A., Bergasa, L. M., Revenga de Toro, P., Nuevo, J., . . . Garrido, M. A. G. (2007). Combination of Feature Extraction Methods for SVM Pedestrian Detection. *Intelligent Transportation Systems, IEEE Transactions on*, 8(2), 292-307. doi: 10.1109/TITS.2007.894194
- Amditis, A., Bimpas, M., Thomaidis, G., Tsogas, M., Netto, M., Mammar, S., . . . Cicilloni, R. (2010). A Situation-Adaptive Lane-Keeping Support System: Overview of the SAFELANE Approach. *IEEE Transactions on Intelligent Transportation Systems*, 11(3), 617-629. doi: 10.1109/TITS.2010.2051667
- Amir, Y., & Wool, A. (1996). *Evaluating quorum systems over the Internet*. Paper presented at the 26th International Symposium on Fault-Tolerant Computing (FTCS-26), Los Alamitos, CA.
- Ammar, H. H., Cukic, B., Mili, A., & Fuhrman, C. (2000). A comparative analysis of hardware and software fault tolerance: Impact on software reliability engineering. *Annals of Software Engineering*, 10, 103-150. doi: Doi 10.1023/A:1018987616443
- Ammoun, S., & Nashashibi, F. (2009). Real time trajectory prediction for collision risk estimation between vehicles. *IEEE Xplore*. doi: 10.1109/ICCP.2009.5284727
- Ansell, J., & Wharton, F. (1992). *Risk: Analysis, Assessment and Management*. Chichester: John Wiley & Sons.
- Antsaklis, P. J., Passino, K. M., & Wang, S. J. (1991). An introduction to autonomous control systems. *IEEE Control Systems*, 11(4), 5-13.

- Apostoloff, N., & Zelinsky, A. (2003, 9-11 June 2003). *Robust vision based lane tracking using multiple cues and particle filtering*. Paper presented at the Intelligent Vehicles Symposium, 2003. Proceedings. IEEE.
- Archer, J. (2005). *Indicators for traffic safety assessment and prediction and their application in micro-simulation modelling : a study of urban and suburban intersections*. (2005:013 Doctoral thesis, monograph), KTH, Stockholm. Retrieved from <http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-143> DiVA database.
- Armand, A., Filliat, D., & Ibañez-Guzman, J. (2014, 8-11 June 2014). *Ontology-based context awareness for driving assistance systems*. Paper presented at the 2014 IEEE Intelligent Vehicles Symposium Proceedings.
- Ashkan Sharafsaleh, & Chan, C.-Y. (November 6-10, 2005). *Experimental Evaluation of Commercially-off-the-Shelf Sensors for Intersection Decision Support Systems*. Paper presented at the 12th World Congress on Intelligent Transportation System, San Francisco. <http://trid.trb.org/view.aspx?id=769544>
- Atev, S., Masoud, O., Janardan, R., & Papanikolopoulos, N. (2005, 2-6 Aug. 2005). *A collision prediction system for traffic intersections*. Paper presented at the Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on.
- Au, S.-k., & L.Beck, J. (2001). Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16(4), 263–277.
- Bahaaldin, K., Fries, R., Bhavsar, P., & Das, P. (2017). A Case Study on the Impacts of Connected Vehicle Technology on No-Notice Evacuation Clearance Time. *Journal of Advanced Transportation*, 2017, 9. doi: 10.1155/2017/6357415
- Bai, C. G. (2005). Bayesian network based software reliability prediction with an operational profile. *Journal of Systems and Software*, 77(2), 103-112.
- Ballard, G. M. (1992). *Industrial risk: safety by design*. Chichester: John Wiley & Sons.

- Basma, F., Tachwali, Y., & Refai, H. H. (2011, 5-7 Oct. 2011). *Intersection collision avoidance system using infrastructure communication*. Paper presented at the Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on.
- Batz, T., Watson, K., & Beyerer, J. (2009). Recognition of dangerous situations within a cooperative group of vehicles. *IEEE Xplore*. doi: 10.1109/IVS.2009.5164400
- Bell, & E, T. (1989). Special Report: Managing Risk in Large Complex Systems. *IEEE Spectrum*, 22-52.
- Bertozzi, M., & Broggi, A. (1998). GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection. *Image Processing, IEEE Transactions on*, 7(1), 62-81. doi: 10.1109/83.650851
- Bertozzi, M., Broggi, A., Chapuis, R., Chausse, F., Fascioli, A., & Tibaldi, A. (2003, 12-15 Oct. 2003). *Shape-based pedestrian detection and localization*. Paper presented at the Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE.
- Bertozzi, M., Broggi, A., Lasagni, A., & Rose, M. D. (2005, 6-8 June 2005). *Infrared stereo vision-based pedestrian detection*. Paper presented at the Intelligent Vehicles Symposium, 2005. Proceedings. IEEE.
- Beucher, S., & Bilodeau, M. (1994, 24-26 Oct. 1994). *Road segmentation and obstacle detection by a fast watershed transformation*. Paper presented at the Intelligent Vehicles '94 Symposium, Proceedings of the.
- Bhavsar, P., Das, P., Paugh, M., Dey, K., & Chowdhury, M. (2017). Risk Analysis of Autonomous Vehicles in Mixed Traffic Streams. *Transportation Research Record: Journal of the Transportation Research Board*, 2625, 51-61. doi: 10.3141/2625-06
- Blanco, M., Atwood, J., Vasquez, H. M., Trimble, T. E., Fitchett, V. L., Radlbeck, & J., M., J. F. (2015). Human Factors Evaluation of Level 2 and Level 3 Automated Driving Concepts. Washington, DC: National Highway Traffic Safety Administration.

- Boehm, B. W. (1991). Software risk management: principles and practices. *IEEE Software*, 8(1), 32 - 41. doi: 10.1109/52.62930
- Bogen, K. T. (1990). *Uncertainty in environmental health risk assessment* New York, NY: Garland Publishing.
- Bose, A., & Ioannou, P. (2001). Evaluation of the environmental effects of intelligent cruise control vehicles. *Transport. Research Record*, 1774, 90–97.
- Box, G. E. P., & Wilson, K. B. (1954). The exploration and exploitation of response surfaces: some general considerations and examples. *Biometrics*, 10, 16-60.
- Broadhurst, A., Baker, S., & Kanade, T. (2005). Monte Carlo road safety reasoning. *IEEE Xplore*. doi: 10.1109/IVS.2005.1505122
- Bucher, C. G., & Bourgund, U. (1990). A Fast and Efficient Response-Surface Approach for Structural Reliability Problems. *Structural Safety*, 7(1), 57-66. doi: Doi 10.1016/0167-4730(90)90012-E
- Buehler, M., Iagnemma, K., & Eds, S. S. (2005). *The 2005 DARPA Grand Challenge*.
- C. Alonso, J. G. (1998). *A method to determine environmental risk in chemical process industries*. Paper presented at the Ninth international symposium loss prevention and safety promotion in the process industries.
- Cafiso, S., Di Graziano, A., & Pappalardo, G. (2013). Using the Delphi method to evaluate opinions of public transport managers on bus safety. *Safety Science*, 57, 254-263. doi: <http://dx.doi.org/10.1016/j.ssci.2013.03.001>
- Calvo-Porrá, C., Faíña-Medín, A., & Nieto-Mengotti, M. (2017). Exploring technology satisfaction: An approach through the flow experience. *Computers in Human Behavior*, 66, 400-408.
- CarSim. (2017). Mechanical Simulation CarSim: Product Overview Retrieved January 5, 2018, from <https://www.carsim.com/products/carsim/index.php>

- Chapman, C. (1997). Project risk analysis and management- PRAM the generic process. *International Journal of Project Management*, 15(5), 273-281.
- Chellappa, R., Gang, Q., & Qinfen, Z. (2004, 17-21 May 2004). *Vehicle detection and tracking using acoustic and video sensors*. Paper presented at the Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on.
- Chen, Z., He, F., Yin, Y., & Du, Y. (2017). Optimal design of autonomous vehicle zones in transportation networks. *Transportation Research Part B: Methodological*, 99, 44-61. doi: <http://dx.doi.org/10.1016/j.trb.2016.12.021>
- Chieh-Chih, W., Thorpe, C., & Suppe, A. (2003). LADAR-based detection and tracking of moving objects from a ground vehicle at high speeds. *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, 416-421. doi: 10.1109/ivs.2003.1212947
- Chowdhury, M., Garber, N., & Li, D. (2000). An interactive multiobjective resource allocation methodology for highway safety improvements. *American Society of Civil Engineers Journal on Infrastructure Systems*.
- Chowdhury, M. A., Garber, N. J., & Li, D. (December 2000). Multiobjective Methodology for Highway Safety Resource Allocation. *Journal of Infrastructure Systems*, 6(4), 138-144.
- Commercial Software for Fault Tree Analysis. United Kingdom: Isograph Software.
- Congress, R. t. (2013). 2013 Status of the Nation's Highways, Bridges, and Transit: Conditions and Performance: United States Department of Transportation.
- D. Godbole, R. Sengupta, J. Misener, N. Kourjanskaia, & J.B. Michael. (January 1998). *Benefits evaluation of crash avoidance systems*. Paper presented at the 77th TRB Meeting, Washington, D.C.
- Davis-McDaniel, C., Chowdhury, M., Pang, W. C., & Dey, K. (2013). Fault-Tree Model for Risk Assessment of Bridge Failure: Case Study for Segmental Box Girder Bridges. *Journal of Infrastructure Systems*, 19(3), 326-334. doi: 10.1061/(Asce)Is.1943-555x.0000129

- Delphi. (2016). Autonomous vehicles annual disengagement report. Retrieved February 28, 2016, from California Department of Autonomous Vehicles [https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/disengagement\\_report](https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/disengagement_report)
- Demartini, J. (2014). The Evolution of Transportation. *Jetset Magazine*.
- Dijkstra, A., Marchesini, P., Bijleveld, F., Kars, V., Drolenga, H., & van Maarseveen, M. (2010). Do Calculated Conflicts in Microsimulation Model Predict Number of Crashes? *Transportation Research Record: Journal of the Transportation Research Board*, 2147, 105-112. doi: 10.3141/2147-13
- Dogan, A., Korkmaz, G., Liu, Y., Ozguner, F., Ozguner, U., Redmill, K., . . . Tokuda, K. (2004, 3-6 Oct. 2004). *Evaluation of intersection collision warning system using an inter-vehicle communication simulator*. Paper presented at the Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on.
- Dupont, S., & Luettin, J. (2000). Audio-visual speech modeling for continuous speech recognition. *IEEE Transactions on Multimedia*, 2(3), 141-151. doi: 10.1109/6046.865479
- Duran, D. R., Robinson, E., Kornecki, A. J., & Zalewski, J. (2013a, September 2013). *Safety Analysis of Autonomous Ground Vehicle Optical Systems: Bayesian Belief Networks Approach*. Paper presented at the Federated Conference on Computer Science and Information Systems, Poland.
- Duran, D. R., Robinson, E., Kornecki, A. J., & Zalewski, J. (2013b, 8-11 Sept. 2013). *Safety analysis of Autonomous Ground Vehicle optical systems: Bayesian belief networks approach*. Paper presented at the Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on.
- Enkelmann, W., Struck, G., & Geisler, J. (1995, 25-26 Sep 1995). *ROMA - a system for model-based analysis of road markings*. Paper presented at the Intelligent Vehicles '95 Symposium., Proceedings of the.



- Eric Fenaux, & Jeremy Buisson. (2007, 2007-6-18 to 2007-6-21). *Enhancement of Car Safety Using Improved ESC and Tyre Development Method*. Paper presented at the 20th International Technical Conference on the Enhanced Safety of Vehicles (ESV), Lyon , France.
- Eriksson, J., Balakrishnan, H., & Madden, S. (2008). *Cabernet: vehicular content delivery using WiFi*. Paper presented at the Proceedings of the 14th ACM international conference on Mobile computing and networking.
- Eskandarian, A., & Thiriez, S. (September 1998). Collision avoidance using a cerebellar model arithmetic computer neural network. *Computer-Aided Civil Infrastructure Engineering*, 13, 303–314.
- Evtimov, I., Eykholt, K., Fernandes, E., Kohno, T., Li, B., Prakash, A., . . . Song, D. (2017). *Robust Physical-World Attacks on Deep Learning Models*. Cornell University Library. Cornell University Library.
- F. Amundsen, & Hyden, C. (1977). Proceedings of first workshop on traffic conflicts: Oslo 77. *Norwegian Council for Scientific and Industrial Research*.
- F. I. Khan, & Abbasi, S. A. (1998). Techniques and methodologies for risk analysis in chemical process industries. *Journal of Loss Prevention in the Process Industries*, 11, 261–277.
- Fagnant, D. J., & Kockelman, K. (2015). Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part a-Policy and Practice*, 77, 167-181. doi: 10.1016/j.tra.2015.04.003
- Fagnant, D. J., & Kockelman, K. (2015). Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, 77(Supplement C), 167-181. doi: <https://doi.org/10.1016/j.tra.2015.04.003>
- Faith Chandler, Addison Heard, Mary Presley, Alexander Burg, Ed Midden, & Mongan, P. (2010). NASA Human Error Analysis (pp. 37): National Aeronautics and Space Administration.

- Fardi, B., Scheunert, U., Cramer, H., & Wanielik, G. (2003, 9-11 June 2003). *A new approach for lane departure identification*. Paper presented at the Intelligent Vehicles Symposium, 2003. Proceedings. IEEE.
- Fardi, B., Schuenert, U., & Wanielik, G. (2005, 6-8 June 2005). *Shape and motion-based pedestrian detection in infrared images: a multi sensor approach*. Paper presented at the Intelligent Vehicles Symposium, 2005. Proceedings. IEEE.
- Fengliang, X., & Fujimura, K. (2002, 17-21 June 2002). *Pedestrian detection and tracking with night vision*. Paper presented at the Intelligent Vehicle Symposium, 2002. IEEE.
- Fennel, H., & Ding, E. L. (2000). A Model-Based Failsafe System for the Continental TEVES Electronic-Stability-Program (ESP). *SAE Technical Paper, 1*. doi: 10.4271/2000-01-1635
- FHWA. (2015). Facts and Statistics - Work Zone Injuries and Fatalities. Retrieved January 20, 2016, from [http://www.ops.fhwa.dot.gov/wz/resources/facts\\_stats/injuries\\_fatalities.htm](http://www.ops.fhwa.dot.gov/wz/resources/facts_stats/injuries_fatalities.htm)
- Fingas, J. (9 May 2015). Self-driving cars can be fooled by fake signals. Retrieved 15 February, 2016, from <http://gizmodo.com/6-simple-things-googles-self-driving-car-still-cant-han-1628040470>
- Fishman, G. S. (1996). *Monte Carlo: concepts, algorithms, and applications*. New York: Springer.
- Franke, U., & Heinrich, S. (2002). Fast obstacle detection for urban traffic situations. *Intelligent Transportation Systems, IEEE Transactions on*, 3(3), 173-181. doi: 10.1109/TITS.2002.802934
- Fuke, Y., & Krotkov, E. (1996, 22-28 Apr 1996). *Dead reckoning for a lunar rover on uneven terrain*. Paper presented at the Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on.

- Furda, A., & Vlacic, L. (2011). Enabling Safe Autonomous Driving in Real-World City Traffic Using Multiple Criteria Decision Making. *IEEE Intelligent Transportation Systems Magazine*, 3(1), 4-17. doi: 10.1109/MITS.2011.940472
- Gadd, S. A., Leeming, D. G., & Riley, T. N. K. (1998). Transport Riskat: The HSE quantified risk assessment tool for toxic and flammable dangerous goods transport by road and rail in Great Britain. *Ninth international symposium loss prevention and safety promotion in the process industries*, 308-317.
- Gavrila, D. M. (2001). Sensor-based pedestrian protection. *Intelligent Systems, IEEE*, 16(6), 77-81. doi: 10.1109/5254.972097
- Gavrila, D. M., Giebel, J., & Munder, S. (2004, 14-17 June 2004). *Vision-based pedestrian detection: the PROTECTOR system*. Paper presented at the Intelligent Vehicles Symposium, 2004 IEEE.
- Geddes, & Bel, N. (1940). *Magic Motorways*. New Work: Random house.
- Gerdes, J., & Hedrick, K. (1997). Vehicle speed and spacing control via coordinated throttle and brake actuation. *Control Engineering Practice*, 5, 1607–1614.
- Germann, S., & Isermann, R. (1995). *Nonlinear distance and cruise control for passenger cars*. Paper presented at the American Control Conference, Seattle, WA.
- Gettman, D., & Head, L. (2003). *Surrogate Safety Measures from Traffic Simulation Models*. (FHWA-RD-03050). McLean, Virginia Federal Highway Administration (FHWA).
- Gettman, D., & Head, L. (2008). *Surrogate safety assessment model and validation: final report*. (FHWA-HRT-08-051). McLean, Virginia: Federal Highway Administration (FHWA).
- Goel, P., Dedeoglu, G., Roumeliotis, S. I., & Sukhatme, G. (2000, 2000). *Fault detection and identification in a mobile robot using multiple model estimation and neural network*. Paper presented at the Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on.

- Gomes, G., May, A., & Horowitz, R. (2004). Calibration of VISSIM for a Congested Freeway (pp. 46). California: California Path Program.
- Google. (2016). Autonomous vehicles annual disengagement report. Retrieved February 28, 2016, from California Department of Autonomous Vehicles [https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/disengagement\\_report](https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/disengagement_report)
- Goyal, A., Lavenberg, S. S., & Trivedi, K. S. (1987). Probabilistic modeling of computer system availability. *Annals of Operations Research Part III: Simulation and Computers*, 8(1), 285-306.
- Greenberg, H. R., & Cramer, J. J. (1991). *Risk Assessment and Risk Management for Chemical Process Industries*. New York: Van Nostrand Reinhold: John Wiley & Sons Inc.
- Gregory J. Skulmoski, F. T. H., Jennifer Krahn, . (2007). The Delphi Method for Graduate Research. *Journal of Information Technology Education*, 6, 1-21.
- H. Dezfuli, A. Benjamin, C. Everett, G. Maggio, M. Stamatelatos, & Youngblood., R. (2011). *NASA Risk Management Handbook*. (NASA/SP-2011-3422). NASA.
- Hany H. Ammar, B. C., Ali Mili, Cris Fuhrman. (November 2000). A comparative analysis of hardware and software fault tolerance: Impact on software reliability engineering. *Annals of Software Engineering*, 10(1), 103-150. doi: 10.1023/A:1018987616443
- Harris, M. (4 Sep 2015). Researcher Hacks Self-driving Car Sensors. Retrieved 15 February, 2016, from <http://spectrum.ieee.org/cars-that-think/transportation/self-driving/researcher-hacks-selfdriving-car-sensors>
- Hasson, F., Keeney, S., & McKenna, H. (2000). Research guidelines for the Delphi survey technique. *Journal of Advanced Nursing*, 32(4), 1008-1015. doi: 10.1046/j.1365-2648.2000.t01-1-01567.x
- Hillenbrand, J., & Kroschel, K. (2006, 7-9 June 2006). *A Study on the Performance of Uncooperative Collision Mitigation Systems at Intersection-like Traffic Situations*. Paper presented at the 2006 IEEE Conference on Cybernetics and Intelligent Systems.

- Hong, C., Nanning, Z., & Junjie, Q. (2005, 6-8 June 2005). *Pedestrian detection using sparse Gabor filter and support vector machine*. Paper presented at the Intelligent Vehicles Symposium, 2005. Proceedings. IEEE.
- Huang, F., Liu, P., Yu, H., & Wang, W. (2013). Identifying if VISSIM simulation model and SSAM provide reasonable estimates for field measured traffic conflicts at signalized intersections. *Accident Analysis & Prevention*, 50, 1014-1024. doi: <https://doi.org/10.1016/j.aap.2012.08.018>
- Hülßen, M., Zöllner, J. M., & Weiss, C. (2011, 5-9 June 2011). *Traffic intersection situation description ontology for advanced driver assistance*. Paper presented at the 2011 IEEE Intelligent Vehicles Symposium (IV).
- Hurst, N. W., Young, S., Donald, I., Gibson, H., & Muyselaar, A. (1996). Measures of safety management performance and attitudes to safety at major hazard sites. *Journal of Loss Prevention in the Process Industr*, 9(2), 161-172.
- J . Richard Kuzmyak, & Dill, J. (2012). Walking and Bicycling in the United States, *TR News*.
- J. D. Lee, M. L. Ries, D. V. McGehee, & Brown, T. L. (2000). *Can Collision Warning Systems Mitigate Distraction Due to In-Vehicle Devices?* . Washington, D.C.: National Highway Traffic Safety Administration.
- J. Tixier, G. Dusserre, O. Salvi, & Gaston, D. (July 2000). Review of 62 risk analysis methodologies of industrial plants. *Journal of Loss Prevention in the Process Industries*, 15(4), 291-303.
- Jihua, H., & Han-Shue, T. (2006, 17-20 Sept. 2006). *Design and implementation of a cooperative collision warning system*. Paper presented at the 2006 IEEE Intelligent Transportation Systems Conference.
- Jocoy, E. H., & Knight, J. R. (1998, 31 Oct-7 Nov 1998). *Adapting radar and tracking technology to an on-board automotive collision warning system*. Paper presented at the 17th DASC. AIAA/IEEE/SAE. Digital Avionics Systems Conference. Proceedings (Cat. No.98CH36267).

- John Hancock, Eric Hoffman, Ryan Sullivan, Darin Ingimarson, Dirk Langer, & Hebert, M. (1997). High-Performance Laser Range Scanner. *SPIE Proceedings on Intelligent Transportation Systems*.
- Jung, C. R., & Kelber, C. R. (2004, 14-17 June 2004). *A lane departure warning system based on a linear-parabolic lane model*. Paper presented at the Intelligent Vehicles Symposium, 2004 IEEE.
- Katrakazas, C., Quddus, M., Chen, W.-H., & Deka, L. (2015). Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, 60, 416-442. doi: <http://dx.doi.org/10.1016/j.trc.2015.09.011>
- Keyarsalan, M., & Montazer, G. A. (2010). INTELLIGENT ONTOLOGICAL AGENT FOR TRAFFIC LIGHT CONTROL OF ISOLATED INTERSECTIONS. *IEEE Xplore*.
- Klein, A. (2016, July 1, 2016). Tesla driver dies in first fatal autonomous car crash in US, *New Scientist*. Retrieved from <https://www.newscientist.com/article/2095740-tesla-driver-dies-in-first-fatal-autonomous-car-crash-in-us/>
- Kuusniemi, H. *User-level reliability and quality monitoring in satellitebased personal navigation*. (Doctor of Technology), Tampere University of Technology, Finland. (544)
- Kuzminski, P., Eisele, J., Garber, N., Schwing, R., Haimes, Y., Li, D., & Chowdhury, M. (1995). Improvement of Highway Safety I: Identification of Causal Factors Through Fault-Tree Modeling. *Risk analysis*, 15(3), 293-312.
- Kyongsu, Y., & Jintai, C. (2001). Nonlinear brake control for vehicle CW/CA systems. *Mechatronics, IEEE/ASME Transactions on*, 6(1), 17-25. doi: 10.1109/3516.914387
- Kyungbok, S., Jae Jun, Y., & Dohyun, K. (2007, Sept. 30 2007-Oct. 3 2007). *Collision Warning System on a Curved Road Using Wireless Sensor Networks*. Paper presented at the Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th.

- Lab, W. B. A Brief History of Autonomous Vehicle Technology. *Wired*.
- Lagadec, P. (1980). *Politique, risque et processus de developpement: Le risque technologique majeur, These de doctorat d'e'tat en science politique*: Pergamon Press.
- Lapp, S. A., & Powers, G. J. (1977). Computer-aided Synthesis of Fault-trees. *Reliability, IEEE Transactions on*, R-26(1), 2-13. doi: 10.1109/TR.1977.5215060
- Laugier, C., Paromtchik, I. E., Perrollaz, M., Yong, M., Yoder, J. D., Tay, C., . . . Nègre, A. (2011). Probabilistic Analysis of Dynamic Scenes and Collision Risks Assessment to Improve Driving Safety. *IEEE Intelligent Transportation Systems Magazine*, 3(4), 4-19. doi: 10.1109/MITS.2011.942779
- Lee, J. (2002). A machine vision system for lane-departure detection. *Computer Vision and Image Understanding*, 86, 52-78.
- Lee, Y. C. (June 1986). *Analysis of range and position comparison methods as a means to provide GPS integrity in the user receiver*. Paper presented at the The 42nd Annual Meeting of The Institute of Navigation, Seattle, Washington.
- Lees, F. P. (1996). *Loss Prevention in the Process Industries: Hazard Identification, Assessment and Control*: Butterworth-Heinemann.
- Li, Y., Yuan, Z., Peng, C., & Lu, S. (2016). *CAP on mobility control for 4G LTE networks*. Paper presented at the Proceedings of the 3rd Workshop on Hot Topics in Wireless, New York City, New York.
- Liang, Z., & Thorpe, C. (1999, 1999). *Stereo- and neural network-based pedestrian detection*. Paper presented at the Intelligent Transportation Systems, 1999. Proceedings. 1999 IEEE/IEEJ/JSAI International Conference on.
- Liangmo Wang, Li Tan, Li-hua An, Zhi-lin Wu, & Li, L. (2012). Study on the ESP System Based on Fuzzy Logic PID Control and Multibody Dynamics, *Journal of Electrical Systems*, 8(1), 57-75.

- Lu, S. (2006). *Vehicle Survivability and Travel Mileage Schedules*. (DOT HS 809 952). Washington, DC: U.S. Department of Transportation Retrieved from <http://www-nrd.nhtsa.dot.gov/Pubs/809952.pdf>.
- Ma, Y., & Trivedi, K. S. (1999). An algorithm for reliability analysis of phased mission systems. *Reliability Engineering and System Safety*, 66, 157-170.
- Mahmassani, H. S., Hou, T., & Dong, J. (2012). Characterizing Travel Time Variability in Vehicular Traffic Networks Deriving a Robust Relation for Reliability Analysis. *Transportation Research Record*, 2315, 141-152.
- Martin, A. (2013). *Interactive Motion Prediction using Game Theory*. (Master of Science), University of Padua, Padova, Italy.
- Mbakwe, A. C., Saka, A. A., Choi, K., & Lee, Y.-J. (2016). Alternative method of highway traffic safety analysis for developing countries using delphi technique and Bayesian network. *Accident Analysis & Prevention*, 93, 135-146. doi: <http://dx.doi.org/10.1016/j.aap.2016.04.020>
- McShane, C. (1995). *Down the Asphalt Path: The Automobile and the American City*: Columbia University Press.
- McShane, C., & Tarr, J. A. (2007). *The Horse in the City: Living Machines in the Nineteenth Century*. Maryland: The Johns Hopkins University Press.
- Memmer, S. (2001). Stability control: Get your yaw-yaws out! , 2015, from <http://www.edmunds.com/ownership/safety/articles/45992/article.html>
- Menon, A., Gorjestani, A., Shankwitz, C., & Donath, M. (2004, 3-6 Oct. 2004). *Roadside range sensors for intersection decision support*. Paper presented at the Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on.
- Mercedes-Benz. (2016). Autonomous vehicles annual disengagement report. Retrieved February 28, 2016, from California Department of Autonomous Vehicles [https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/disengagement\\_report](https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/disengagement_report)



- Meyer-Delius, D., Plagemann, C., & Burgard, W. (2009). Probabilistic Situation Recognition for Vehicular Traffic Scenarios. *2009 IEEE International Conference on Robotics and Automation*.
- Michael Sivak, & Schoettle, B. (2015). Influence of Current Nondrivers on the Amount of Travel and Trip Patterns with Self-Driving Vehicles. Michigan: University of Michigan Transportation Research Institute
- Miller, R., & Qingfeng, H. (2002, 2002). *An adaptive peer-to-peer collision warning system*. Paper presented at the Vehicular Technology Conference. IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No.02CH37367).
- Mohammad, M. A., Kaloskampis, I., Hicks, Y., & Setchi, R. (2015). Ontology-based Framework for Risk Assessment in Road Scenes Using Videos. *Procedia Computer Science*, 60, 1532-1541. doi: <http://dx.doi.org/10.1016/j.procs.2015.08.300>
- Mohan, A., Papageorgiou, C., & Poggio, T. (2001). Example-based object detection in images by components. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(4), 349-361. doi: 10.1109/34.917571
- Nanda, H., & Davis, L. (2002, 17-21 June 2002). *Probabilistic template based pedestrian detection in infrared videos*. Paper presented at the Intelligent Vehicle Symposium, 2002. IEEE.
- . National Motor Vehicle Crash Causation Survey. (July 2008). Virginia: National Highway Traffic Safety Administration.
- NHTSA. (2015). *Traffic Safety Facts 2013: A Compilation of Motor Vehicle Crash Data from the Fatality Analysis Reporting System and the General Estimates System*. (DOT HS 812 139). Washington, D.C.: National Highway Traffic Safety Administration Retrieved from <https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812139>.
- NHTSA. (2016a). *Early Estimate of Motor vehicle Traffic*. (DOTHS812332). Washington, DC: National Highway Traffic Safety Administration.

- NHTSA. (2016b). Fatality Analysis Reporting System (FARS) Encyclopedia. Retrieved June 20, 2017, from National Highway Traffic Safety Administration (NHTSA)
- NHTSA. (2016c). *Pedestrians: 2014 data (Traffic Safety Facts)*. (DOT HS 812 270). Washington, DC: National Center for Statistics and Analysis, National Highway Traffic Safety Administration.
- NHTSA. (2016d). *Traffic Safety Facts 2014: A Compilation of Motor Vehicle Crash Data from the Fatality Analysis Reporting System and the General Estimates System*. (DOT HS 812 261). Washington, D.C.: National Highway Traffic Safety Administration Retrieved from <https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812261>.
- Nissan. (2016). Autonomous vehicles annual disengagement report. Retrieved February 28, 2016, from California Department of Autonomous Vehicles [https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/disengagement\\_report](https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/disengagement_report)
- Norsys. Netica User Manual. 2017, from <http://www.norsys.com/index.html>
- Nothdurft, T., Hecker, P., Ohl, S., Saust, F., Maurer, M., Reschka, A., & Böhmer, J. R. (2011, 5-7 Oct. 2011). *Stadt-pilot: First fully autonomous test drives in urban traffic*. Paper presented at the 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC).
- Nowakowski, C., Shladover, S. E., Chan, C.-Y., & Tan, H.-S. (2015). Development of California Regulations to Govern Testing and Operation of Automated Driving Systems. *Transportation Research Record: Journal of the Transportation Research Board*, 2489, 137-144. doi: 10.3141/2489-16
- NYSDOT. (2015). Summary of Motor Vehicle Crashes: 2014 Statewide Statistical Summary. Retrieved February 15, 2016, from New York State Department of Motor Vehicles <https://dmv.ny.gov/statistic/2014-nyscrashsummary.pdf>
- ODOT. (2011). *Protocol for Vissim Simulation* Oregon: Oregon Department of Transportation Retrieved from <http://www.oregon.gov/ODOT/Planning/Pages/APM.aspx>.

- Onieva, E., Alonso, J., Perez, J., Milanés, V., & de Pedro, T. (2009, 20-24 Aug. 2009). *Autonomous car fuzzy control modeled by iterative genetic algorithms*. Paper presented at the Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on.
- P. Ryus, M. V., L. Elefteriadou, R. G. Dowling, & Ostrom, B. K. (2011). Highway capacity manual 2010. *TR News*, 273, 45–48.
- Pel, A., Bliemer, M., & Hoogendoorn, S. (2011). Modelling Traveller Behaviour under Emergency Evacuation Conditions (pp. 166-193). Delft: EJTIR.
- PennDOT. (2016). *Pennsylvania Highway Statistics: 2015 Highway Data*. Pennsylvania Department of Transportation Retrieved from <http://www.penndot.gov/ProjectAndPrograms/Planning/TrafficInformation/Pages/2015-Highway-Statistics.aspx>.
- Persson, M., Botling, F., Hesslow, E., & Johansson, R. (1999, 1999). *Stop and go controller for adaptive cruise control*. Paper presented at the Control Applications, 1999. Proceedings of the 1999 IEEE International Conference on.
- Petridou, E., & Moustaki, M. (2000). Human Factors in the Causation of Road Traffic Crashes. *European Journal of Epidemiology*, 16(9), 819-826.
- Pinto, C. (2012). How autonomous vehicle policy in California and Nevada addresses technological and non-technological liabilities. *Intersect: The Stanford Journal of Science, Technology and Society*, 5.
- Platho, M., Groß, H. M., & Eggert, J. (2013, 23-26 June 2013). *Predicting Velocity Profiles of Road Users at Intersections Using Configurations*. Paper presented at the 2013 IEEE Intelligent Vehicles Symposium (IV).
- Pollard, E., Morignot, P., & Nashashibi, F. (2013, 9-12 July 2013). *An ontology-based model to determine the automation level of an automated vehicle for co-driving*. Paper presented at the Proceedings of the 16th International Conference on Information Fusion.

- Pomerleau, D. (1995, 25-26 Sep 1995). *RALPH: rapidly adapting lateral position handler*. Paper presented at the Intelligent Vehicles '95 Symposium., Proceedings of the.
- Pommerening, F., Wölfl, S., & Westphal, M. (2009). Right-of-Way Rules as Use Case for Integrating GOLOG and Qualitative Reasoning. *Advances in Artificial Intelligence*, 5803, 468-475.
- Ponsa, D., Lopez, A., Lumbreras, F., Serrat, J., & Graf, T. (2005, 13-15 Sept. 2005). *3D vehicle sensor based on monocular vision*. Paper presented at the Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE.
- Porter, C. E. (1981). *Risk Allowance in Construction Contracts* (M.Sc), University of Manchester.
- PTVGroup. (2015). PTV Vissim Version 8: PTV Group.
- Pu, L., & Joshi, R. (2008). Surrogate Safety Assessment Model (SSAM): Software User Manual (pp. 96). McLean, Virginia: Federal Highway Administration.
- Qingyou, L., & Hao, C. (1999). Failure analysis for the casing *Journal of Southwest Petroleum Institute*, 21(4), 75–77.
- Rauzy, A. (1993). New algorithms for fault trees analysis. *Reliability Engineering & System Safety*, 40(3), 203-211.
- Rawashdeh, Z. Y., & Mahmud, S. M. (2008, 10-12 Jan. 2008). *Intersection Collision Avoidance System Architecture*. Paper presented at the Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE.
- . Report to Congress on the National Highway Traffic Safety Administration ITS Program. (January 1997): United States Department of Transportation.
- Richard Wallace, & Silberg, G. (August 2012). Self-driving cars: The next revolution: Center for Automotive Research, Transportation Systems Analysis Group.

- Rieger, W. G. (1986). Directions in Delphi developments: Dissertations and their quality. *Technological Forecasting and Social Change*, 29(2), 195-204. doi: [http://dx.doi.org/10.1016/0040-1625\(86\)90063-6](http://dx.doi.org/10.1016/0040-1625(86)90063-6)
- Risack, R., Mohler, N., & Enkelmann, W. (2000, 2000). *A video-based lane keeping assistant*. Paper presented at the Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE.
- Roenker, D. L., Cissell, G. M., Ball, K. K., Wadley, V. G., & Edwards, J. D. (2003). Speed-of-Processing and Driving Simulator Training Result in Improved Driving Performance. *Human Factors*, 45(2), 218-233. doi: 10.1518/hfes.45.2.218.27241
- Ross, T., & Donald, S. (1995). A fuzzy multi-objective approach to risk management. *Computing in civil engineering: 2nd congress held in conjunction with A/E/C systems 95*, 95(2), 1400- 1403.
- Rossetti, M. D., & Ni, Q. (2010). *Simulating Large-Scale Evacuation Scenarios in Commercial Shopping Districts*. Paper presented at the Proceedings of the 2010 Winter Simulation Conference, Baltimore.
- Rowe, G., & Wright, G. (2001). Expert Opinions in Forecasting: The Role of the Delphi Technique. In J. S. Armstrong (Ed.), *Principles of Forecasting: A Handbook for Researchers and Practitioners* (pp. 125-144). Boston, MA: Springer US.
- S. Park, T. Kim, S. Kang, & Heon, K. (2003). A Novel Signal Processing Technique for Vehicle Detection Radar. *2003 IEEE MTT-S Int'l Microwave Symposium Digest*, 607-610 doi: 10.1109/MWSYM.2003.1211012
- S. Yacoub, H. A., and T. Robinson, , pp. 50-61, Nov. . (November 1999). Dynamic Metrics for Object Oriented Designs. *Proceeding Sixth International Symposium Software Metrics*, 99, 50-61.
- Safie, F. M., Stutts, R. G., & Huang, Z. (2015, 26-29 Jan. 2015). *Reliability and probabilistic risk assessment &#x2014; How they play together*. Paper presented at the 2015 Annual Reliability and Maintainability Symposium (RAMS).

- Salomon, I., & Mokhtarian, P. L. (1997). Coping with congestion: Understanding the gap between policy assumptions and behavior. *Transportation Research Part D: Transport and Environment*, 2(2), 107-123. doi: [http://dx.doi.org/10.1016/S1361-9209\(97\)00003-5](http://dx.doi.org/10.1016/S1361-9209(97)00003-5)
- Santos, A., McGuckin, N., Nakamoto, H. Y., Gray, D., & Liss, S. (2011). *Summary of Travel Trends: 2009 National Household Travel Survey*. (FHWA-PL-11-022). Washington, DC Federal Highway Administration.
- Schroeder, P., & Wilbur, M. (2013). *2012 National survey of bicyclist and pedestrian attitudes and behavior, volume 1: Summary report*. (DOT HS 811 841). Washington, DC: National Highway Traffic Safety Administration.
- Seiler, P., Song, B., & Hedrick, K. (1998). Development of a Collision Avoidance System. *SAE International*.
- Shibahata, Y., Shimada, K., & Tomari, T. (1993). Improvement of Vehicle Maneuverability by Direct Yaw Moment Control. *Vehicle System Dynamics*, 22(5-6), 465-481. doi: Doi 10.1080/00423119308969044
- Simonite, T. (2016, January 26, 2016). Your Future Self-Driving Car Will Be Way More Hackable. *MIT Technology Review*.
- Smith, B. W. (2012). Managing Autonomous Transportation Demand. *Santa Clara Law Review*, 52(4), 22.
- Sorokanich, R. (30 August 2014). 6 Simple Things Google's Self-Driving Car Still Can't Handle. Retrieved 15 February, 2016, from <http://www.engadget.com/2015/09/05/self-driving-car-lidar-exploit>
- Stefan Trommer, Viktoriya Kolarova, Eva Fraedrich, Lars Kröger, Benjamin Kickhöfer, Tobias Kuhnimhof, . . . Phleps, P. (2016). Autonomous Driving - The Impact of Vehicle Automation on Mobility Behaviour (pp. 29). Germany Institute of Mobility Research
- Stiller, C., Kammel, S., & Lulcheva, I. (2008). Probabilistische Inferenz für Kognitive Automobile. *ANWENDUNGEN*.

- Suard, F., Guigue, V., Rakotomamonjy, A., & Benshrir, A. (2005, 6-8 June 2005). *Pedestrian detection using stereo-vision and graph kernels*. Paper presented at the Intelligent Vehicles Symposium, 2005. Proceedings. IEEE.
- Sun, Z., Bebis, G., & Miller, R. (2006). On-road vehicle detection: a review. *IEEE Trans Pattern Anal Mach Intell*, 28(5), 694-711. doi: 10.1109/TPAMI.2006.104
- Swaroop, D., & Huandra, R. (1998). Intelligent cruise control design based on a traffic flow specification. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, 30, 319-344.
- Swaropp, D., & Rajagopal, K. (1999). Intelligent cruise control systems and traffic flow stability. *Transportation Research Part C*, 7, 329–352.
- Swarup, M. B., & Rao, M. S. (2014). Safety Analysis of Adaptive Cruise Control System Using FMEA and FTA *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(6), 330-337.
- Swerling, P. (1997). Radar probability of detection for some additional fluctuating target cases. *IEEE Transactions on aerospace and electronic systems*, 33(2), 698-709. doi: Doi 10.1109/7.588492
- Szarvas, M., Yoshizawa, A., Yamamoto, M., & Ogata, J. (2005, 6-8 June 2005). *Pedestrian detection with convolutional neural networks*. Paper presented at the Intelligent Vehicles Symposium, 2005. Proceedings. IEEE.
- T. Chira-Chavala, & S.M. Yoo. (1994). Potential safety benefits of intelligent cruise control systems. *Accident Analysis and Prevention*, 26(2), 135–146.
- Tarr, J. A. (1996). *The search for the ultimate sink: urban pollution in historical perspective*. Ohio: University of Akron press.
- Tiemessen, G., & Zweeden, J. P. v. (1998). Risk assessment of the transport of hazardous materials. *Ninth international symposium loss prevention and safety promotion in the process industries*, 299-307.

- Tingvall, C., Krafft, M., Kullgren, A., & Lie, A. (May 19-22, 2003). The effectiveness of ESP (Electronic Stability Programme) in reducing real life accidents. *Traffic Injury Prevention*, 5, 37-41.
- Touran, A., Brackstone, M., & McDonald, M. (May 1999). A collision model for safety evaluation of autonomous intelligent cruise control. *Accident Analysis and Prevention*, 31, 567-578.
- Tupper, L. L., Chowdhury, M., & Sharp, J. (2014). *Tort liability risk prioritization through the use of fault tree analysis*. Paper presented at the Proceedings of Transportation Research Board 93rd Annual Meeting.
- Turnbull, K. F. (2015, November 4-5, 2015). *Automated and Connected Vehicles: Summary of the 9th University Transportation Centers Spotlight Conference*. Paper presented at the 9th University Transportation Centers Spotlight Conference, Washington, D.C.
- Ullman, G. L., Finley, M. D., Bryden, J. E., Srinivasan, R., & Council, F. M. (2008). Traffic Safety Evaluation of Nighttime and Daytime Work Zones *National cooperative highway research program* (pp. 89). Washington, D.C.: Transportation Research Board.
- Vacek, S., Gindele, T., Zollner, J. M., & Dillmann, R. (2007, 13-15 June 2007). *Situation classification for cognitive automobiles using case-based reasoning*. Paper presented at the 2007 IEEE Intelligent Vehicles Symposium.
- van Zanten, A. T., Erhardt, R., & Pfaff, G. (1995). VDC, The Vehicle Dynamics Control System of Bosch. *SAE Technical Paper*, 1. doi: 10.4271/950759
- VanderWerf, J., Shladover, S. E., Miller, M. A., & Kourjanskaia, N. (January 2002). Effects of adaptive cruise control systems on highway traffic flow capacity. *Transp. Research Record*, 1800, 78-84.
- Vasconcelos, L., Neto, L., Seco, Á., & Silva, A. (2014). Validation of the Surrogate Safety Assessment Model for Assessment of Intersection Safety. *Transportation Research Record: Journal of the Transportation Research Board*, 2432, 1-9. doi: 10.3141/2432-01



- Vattulainen, I., & Ala-Nissila, T. (1995). Mission Impossible: Find a Random Pseudorandom Number Generator. *Computers in Physics*, 9(5), 500-510. doi: 10.1063/1.168548
- VDOT. (2015). Virginia Traffic Crash Facts 2014. Retrieved February 15, 2016, from Virginia Highway Safety Office, Virginia Department of Motor Vehicles [http://www.dmv.state.va.us/safety/#crash\\_data/crash\\_facts/index.asp](http://www.dmv.state.va.us/safety/#crash_data/crash_facts/index.asp)
- Vesely, W. E. (1984). Robust Risk Analysis: The Need for it in Nuclear Probabilistic Risk Evaluations. In R. A. Waller & V. T. Covello (Eds.), *Low-Probability High-Consequence Risk Analysis: Issues, Methods, and Case Studies* (pp. 153-160). Boston, MA: Springer US.
- Vesely, W. E., Goldberg, F. F., Roberts, N. H., & Haasl, D. F. (1981). *Fault Tree Handbook*. Washington, D.C.
- Vincent, J. (13 January 2016). Google's self-driving cars would've hit something 13 times if not for humans. Retrieved 15 February, 2016, from <http://www.theverge.com/2016/1/13/10759424/google-self-driving-car-accidents-driver-disengagements>
- . VISSIM 7 User Manual. (2015). Karlsruhe, Germany.
- Vogt, A. (1999). *Crash Models for Rural Intersections: Four-Lane by Two-Lane Stop Controlled and Two-Lane by Two-Lane Signalized*. (FHWA-RD-99-128). Washington, D.C.: Federal Highway Administration (FHWA).
- Volkanovski, A., Cepin, M., & Mavko, B. (2009). Application of the fault tree analysis for assessment of power system reliability. *Reliability Engineering & System Safety*, 94(6), 1116-1127. doi: 10.1016/j.res.2009.01.004
- Volkswagen. (2016). Autonomous vehicles annual disengagement report. Retrieved February 28, 2016, from California Department of Autonomous Vehicles [https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/disengagement\\_report](https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/disengagement_report)

- W. P. G. Schlechter. (1996). Facility risk review as a mean to addressing existing risks during the life cycle of a process unit, operation or facility. *International Journal of Pressure Vessels and Piping*, 66(1-3), 387–402.
- W.G. Najm, & A.L. Burgett. (1997). *Benefits estimation for selected collision avoidance systems*. Paper presented at the The 4th World Congress on Intelligent Transport Systems, Berlin.
- W.G. Najm, J.D. Smith, & M. Yanagisawa. (2007). Pre-Crash Scenario Typology for Crash Avoidance Research. Washington, D.C.: National Highway Traffic Safety Administration.
- Wang, H., & Xue, C. (2004). Modelling and Simulation of Electric Stability Program for the Passenger Car. *SAE Technical Paper*, 1. doi: 10.4271/2004-01-2090
- Webots. (2017). Robot Simulator Webot: Product Overview. Retrieved January 5, 2018, 2018, from <https://www.cyberbotics.com/webots.php>
- White, A. V., & Burton, I. (1980). *Environmental Risk Assessment*: John Wiley & Sons.
- White, D. (1995). Application of systems thinking to risk management: A review of the literature. *Journal of Management History (Archive) merged into Management Decision*, 33(10), 35-45. doi: <http://dx.doi.org/10.1108/EUM0000000003918>
- Wilson, & H. C & Keller, A. Z. (1990). *Disaster prevention, planning and limitation*. London: British Library.
- Wilson, B. (2001). *How soon to brake and how hard to brake: Unified analysis of the envelope of opportunity for rear-end collision warning*. Paper presented at the International Technical Conference Enhanced Safety of Vehicles, Amsterdam, Netherlands.
- Wo, x, hler, C., & Anlauf, J. K. (1999). An adaptable time-delay neural-network algorithm for image sequence analysis. *Neural Networks, IEEE Transactions on*, 10(6), 1531-1536. doi: 10.1109/72.809100

- Woll, J. (1997). Radar Based Adaptive Cruise Control for Truck Applications. *SAE International*. doi: 10.4271/973184
- Worrall, S., Orchansky, D., Masson, F., & Nebot, E. (2010). Improving Vehicle Safety Using Context Based Detection of Risk. *Annual Conference on Intelligent Transportation Systems*.
- WSDOT. (2014). *Protocol for VISSIM Simulation*. Washinton: Washington State Department of Transportation.
- X. Lu, H. T., Shladover, S., & Hedrick, K. (June 2001). Nonlinear longitudinal controller implementation and comparison for automated cars. *Journal of Dynamic Systems, Measurement, and Control*, 123, 161–167.
- Y. Jia, J. Song, & Sun, Q. (2004). Dynamic Model and Control Method for Motor Vehicle with Electronic Stability Program. *Journal of Highway and Transportation Research and Development*, 21(5), 132-136.
- Yi, K., Woo, M., Kim, S., & Lee, S. (March 1999). Study on a road-adaptive CW/CA algorithm for automobiles using HiL simulations. *JSME International Journal Series C*, 42, 163–170.
- Yllera, J. (1988). *Engineering Risk and Hazard Assessment (Kandel and Avni, Eds.)*. Florida Florida: CRC Press Inc.
- Yu, X., Beucher, S., & Bilodeau, M. (1992, 29 Jun-1 Jul 1992). *Road tracking, lane segmentation and obstacle recognition by mathematical morphology*. Paper presented at the Intelligent Vehicles '92 Symposium., Proceedings of the.
- Zehang, S., Bebis, G., & Miller, R. (2002, 2002). *On-road vehicle detection using Gabor filters and support vector machines*. Paper presented at the Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on.

## Appendix A

### Calculation of Simulation Runs Number

Table 13

*Calculation based on network average speed for time interval 600-1500 seconds*

Simulation Runs	Random Seed	Time Interval (sec)	Average Speed (mph)
1	5	600 – 1500	31.25272
2	10	600 – 1500	31.63295
3	15	600 – 1500	29.05456
4	20	600 – 1500	31.14431
5	25	600 – 1500	30.92869
6	30	600 – 1500	30.1724
7	35	600 – 1500	30.14788
8	40	600 – 1500	31.44893
9	45	600 – 1500	29.95752
10	50	600 – 1500	30.89594
Average			30.66359
Standard Deviation			0.806473

$$N = \left( 2 \times 2.2622 \times \frac{0.8065}{0.10 \times 30.6636} \right)^2 = 1.42 \text{ runs}$$

Table 14

*Calculation based on network average speed for time interval 1500-2400 seconds*

Simulation Runs	Random Seed	Time Interval (sec)	Average Speed (mph)
1	5	1500 – 2400	27.46423
2	10	1500 – 2400	26.8748
3	15	1500 – 2400	25.69973
4	20	1500 – 2400	25.69288
5	25	1500 – 2400	25.37566

Table 14 (continued)

Simulation Runs	Random Seed	Time Interval (sec)	Average Speed (mph)
6	30	1500 – 2400	26.74075
7	35	1500 – 2400	25.29741
8	40	1500 – 2400	26.50471
9	45	1500 – 2400	26.59292
10	50	1500 – 2400	25.47071
Average			26.17138
Standard Deviation			0.754143

$$N = \left( 2 \times 2.2622 \times \frac{0.7541}{0.10 \times 26.1714} \right)^2 = 1.70 \text{ runs}$$

Table 15

*Calculation based on network average speed for time interval 2400-3300 seconds*

Simulation Runs	Random Seed	Time Interval (sec)	Average Speed (mph)
1	5	2400 – 3300	24.44064
2	10	2400 – 3300	24.02435
3	15	2400 – 3300	23.40577
4	20	2400 – 3300	22.74352
5	25	2400 – 3300	22.16012
6	30	2400 – 3300	24.02695
7	35	2400 – 3300	22.17685
8	40	2400 – 3300	23.42267
9	45	2400 – 3300	23.97453
10	50	2400 – 3300	22.39201
Average			23.27674
Standard Deviation			0.851464

$$N = \left( 2 \times 2.2622 \times \frac{0.8515}{0.10 \times 23.2767} \right)^2 = 2.74 \text{ runs}$$

Table 16

*Calculation based on network average speed for time interval 3300-4200 seconds*

Simulation Runs	Random Seed	Time Interval (sec)	Average Speed (mph)
1	5	3300 – 4200	21.74673
2	10	3300 – 4200	22.10281
3	15	3300 – 4200	21.24155
4	20	3300 – 4200	20.48982
5	25	3300 – 4200	19.77108
6	30	3300 – 4200	21.91904
7	35	3300 – 4200	19.96118
8	40	3300 – 4200	21.25557
9	45	3300 – 4200	21.67305
10	50	3300 – 4200	20.07585
Average			21.02367
Standard Deviation			0.875165

$$N = \left( 2 \times 2.2622 \times \frac{0.8752}{0.10 \times 21.0237} \right)^2 = 3.55 \text{ runs}$$

Table 17

*Calculation based on average travel time for different time interval for travel time measurement segment 1*

Simulation Runs	Random Seed	Average Travel Time (sec) for time interval 600- 1500 secs	Average Travel Time (sec) for time interval 1500- 2400 secs	Average Travel Time (sec) for time interval 2400- 3300 secs	Average Travel Time (sec) for time interval 3300- 4200 secs
1	5	289.32	300.07	304.58	333.91
2	10	304.57	322.29	319.85	351.96
3	15	316.32	335.71	333.11	360.21
4	20	295.44	321.49	328.47	342.9
5	25	322.43	343.74	355.64	387.65
6	30	337.17	349.18	343.59	386.76
7	35	311.58	328.2	334.11	367.88
8	40	293.79	307.55	314.11	331.61
9	45	323.42	346.82	334.58	391.14

Table 17 (continued)

Simulation Runs	Random Seed	Average Travel Time (sec) for time interval 600-1500 secs	Average Travel Time (sec) for time interval 1500-2400 secs	Average Travel Time (sec) for time interval 2400-3300 secs	Average Travel Time (sec) for time interval 3300-4200 secs
10	50	328.51	354.56	368.89	397.45
Average		312.255	330.961	333.693	365.147
SD		16.12081	18.26060	19.06049	24.70667
Number of Simulation Runs		5.46	6.23	6.68	9.37

## Appendix B

### External Driver Model Code (DLL File Development)

```
/* June. 2017 */

/* Autonomous Platooning for MS Thesis Work */

/* Modified by Plaban Das, Rowan University. */

#include "DriverModel.h"

/*=====
=====*/

/* These values are placeholders and declarations. */

/*===== Current Vehicle =====*/

double time_step = 0.0;

long current_vehID = 0;

long current_lane = 0;

double current_lateral_pos = 0.0;

double current_speed = 0.0;

double current_accerleration = 0.0;

double current_length = 0.0;

double max_acceleration = 0.0;

long turning_indicator = 0;

long current_category = 0;

double desired_velocity = 0.0;

double current_type = 0.0;

long vehicle_color = RGB(0,0,0);
```



```

/*===== Lead Vehicle =====*/

long lead_vehID = 0;

double lead_vehicle_lateral_position = 0.0;

double lead_vehicle_distance = 0.0;

double lead_vehicle_speed_difference = 0.0;

double lead_vehicle_acceleration = 0.0;

double lead_vehicle_length = 0.0;

long lead_vehicle_category = 0;

/*===== Desired =====*/

double desired_speed_limit = 0.0;

double desired_acceleration = 0.0;

double desired_lane_angle = 0.0;

long active_lane_change = 0;

long rel_target_lane = 0;

/*=====

=====*/

BOOL APIENTRY DllMain (HANDLE hModule,

                      DWORD ul_reason_for_call,

                      LPVOID lpReserved)

{

    switch (ul_reason_for_call) {

        case DLL_PROCESS_ATTACH:

        case DLL_THREAD_ATTACH:

```

```

    case DLL_THREAD_DETACH:

    case DLL_PROCESS_DETACH:

        break;

}

return TRUE;

}

/*=====
=====*/

DRIVERMODEL_API int DriverModelSetValue (long  type,

                                         long  index1,

                                         long  index2,

                                         long  long_value,

                                         double double_value,

                                         char  *string_value)

{

    /* Sets the value of a data object of type <type>, selected by <index1> */

    /* and possibly <index2>, to <long_value>, <double_value> or          */

    /* <*string_value> (object and value selection depending on <type>).  */

    /* Return value is 1 on success, otherwise 0.                        */

    switch (type) {

        case DRIVER_DATA_PATH          :

        case DRIVER_DATA_TIMESTEP      :

```

```

    return 1;

case DRIVER_DATA_TIME          :

    time_step = double_value;

    return 1;

case DRIVER_DATA_VEH_ID        :

    /* reset leading vehicle's data for this new vehicle */

    current_vehID = long_value;

    /* lead_vehicle_distance      = 999.0;

    lead_vehicle_speed_difference = -99.0;

    lead_vehicle_length          = 0.0; */

    return 1;

case DRIVER_DATA_VEH_LANE      :

    current_lane = long_value;

    return 1;

case DRIVER_DATA_VEH_ODOMETER  :

case DRIVER_DATA_VEH_LANE_ANGLE :

case DRIVER_DATA_VEH_LATERAL_POSITION :

    current_lateral_pos = double_value;

    return 1;

case DRIVER_DATA_VEH_VELOCITY  :

    /* current vehicle velocity */

    current_speed = double_value;

    return 1;

```

```

case DRIVER_DATA_VEH_ACCELERATION      :

    /* vehicle's current acceleration */

    current_accerleration = double_value;

    return 1;

case DRIVER_DATA_VEH_LENGTH            :

    /* vehicle's current length */

    current_length = double_value;

    return 1;

case DRIVER_DATA_VEH_WIDTH              :

case DRIVER_DATA_VEH_WEIGHT             :

case DRIVER_DATA_VEH_MAX_ACCELERATION   :

    /* vehicle's current maximum acceleration */

    max_acceleration = double_value;

    return 1;

case DRIVER_DATA_VEH_TURNING_INDICATOR :

    turning_indicator = long_value;

    return 1;

case DRIVER_DATA_VEH_CATEGORY           :

    /* vehicle's category */

    current_category = long_value;

    return 1;

case DRIVER_DATA_VEH_PREFERRED_REL_LANE :

case DRIVER_DATA_VEH_USE_PREFERRED_LANE :

```

```

    return 1;

case DRIVER_DATA_VEH_DESIRED_VELOCITY :

    /* vehicle's desired velocity */

    desired_velocity = double_value;

    return 1;

case DRIVER_DATA_VEH_X_COORDINATE :

case DRIVER_DATA_VEH_Y_COORDINATE :

case DRIVER_DATA_VEH_TYPE :

    /* vehicle's current type */

    current_type = double_value;

    return 1;

case DRIVER_DATA_VEH_COLOR :

    vehicle_color = long_value;

    return 1;

case DRIVER_DATA_VEH_CURRENT_LINK :

    return 0; /* (To avoid getting sent lots of DRIVER_DATA_VEH_NEXT_LINKS
messages) */

    /* Must return 1 if these messages are to be sent from VISSIM! */

case DRIVER_DATA_VEH_NEXT_LINKS :

case DRIVER_DATA_VEH_ACTIVE_LANE_CHANGE :

case DRIVER_DATA_VEH_REL_TARGET_LANE :

case DRIVER_DATA_NVEH_ID :

    /* lead vehicle's ID */

```

```

    if (index1 == 0 && index2 == 1) {

        lead_vehID = long_value;

    }

    return 1;

case DRIVER_DATA_NVEH_LANE_ANGLE      :

case DRIVER_DATA_NVEH_LATERAL_POSITION :

    /* lead vehicle's lateral position */

    if (index1 == 0 && index2 == 1){/* leading vehicle on the same lane as following
vehicle */

        lead_vehicle_lateral_position = double_value;

    }

    return 1;

case DRIVER_DATA_NVEH_DISTANCE        :

    /* lead vehicle's distance */

    if (index1 == 0 && index2 == 1) { /* leading vehicle on own lane */

        lead_vehicle_distance = double_value;

    }

    return 1;

case DRIVER_DATA_NVEH_REL_VELOCITY    :

    if (index1 == 0 && index2 == 1) { /* leading vehicle on own lane */

        lead_vehicle_speed_difference = double_value;

    }

    return 1;

```

```

case DRIVER_DATA_NVEH_ACCELERATION    :

    if (index1 == 0 && index2 == 1) { /* leading vehicle on own lane */

        lead_vehicle_acceleration = double_value;

    }

    return 1;

case DRIVER_DATA_NVEH_LENGTH          :

    if (index1 == 0 && index2 == 1) { /* leading vehicle on own lane */

        lead_vehicle_length = double_value;

    }

    return 1;

case DRIVER_DATA_NVEH_WIDTH           :

    return 1;

case DRIVER_DATA_NVEH_WEIGHT          :

case DRIVER_DATA_NVEH_TURNING_INDICATOR :

case DRIVER_DATA_NVEH_CATEGORY        :

    if (index1 == 0 && index2 == 1) { /* leading vehicle on own lane */

        lead_vehicle_category = long_value;

    }

    return 1;

case DRIVER_DATA_NVEH_LANE_CHANGE     :

case DRIVER_DATA_NO_OF_LANES          :

case DRIVER_DATA_LANE_WIDTH           :

case DRIVER_DATA_LANE_END_DISTANCE    :

```

```

case DRIVER_DATA_RADIUS          :

case DRIVER_DATA_MIN_RADIUS      :

case DRIVER_DATA_DIST_TO_MIN_RADIUS :

case DRIVER_DATA_SLOPE          :

case DRIVER_DATA_SLOPE_AHEAD     :

case DRIVER_DATA_SIGNAL_DISTANCE :

case DRIVER_DATA_SIGNAL_STATE    :

case DRIVER_DATA_SIGNAL_STATE_START :

case DRIVER_DATA_SPEED_LIMIT_DISTANCE :

case DRIVER_DATA_SPEED_LIMIT_VALUE  :

    desired_speed_limit = double_value;

    return 1;

case DRIVER_DATA_DESIRED_ACCELERATION :

    desired_acceleration = double_value;

    return 1;

case DRIVER_DATA_DESIRED_LANE_ANGLE :

    desired_lane_angle = double_value;

    return 1;

case DRIVER_DATA_ACTIVE_LANE_CHANGE :

    active_lane_change = long_value;

    return 1;

case DRIVER_DATA_REL_TARGET_LANE :

    rel_target_lane = long_value;

```



```

    return 1;

default :

    return 0;

}

}

/*-----*/

DRIVERMODEL_API int DriverModelGetValue (long  type,

                                         long  index1,

                                         long  index2,

                                         long  *long_value,

                                         double *double_value,

                                         char  **string_value)

{

    /* Gets the value of a data object of type <type>, selected by <index1> */

    /* and possibly <index2>, and writes that value to <*double_value>,    */

    /* <*float_value> or <**string_value> (object and value selection      */

    /* depending on <type>).                                              */

    /* Return value is 1 on success, otherwise 0.                        */

    switch (type) {

        case DRIVER_DATA_STATUS :

            *long_value = 0;

            return 1;

        case DRIVER_DATA_VEH_TURNING_INDICATOR :

```

```

    *long_value = turning_indicator;

    return 1;

case DRIVER_DATA_VEH_DESIRED_VELOCITY :

    *double_value = desired_velocity;

    return 1;

case DRIVER_DATA_VEH_COLOR :

    *long_value = vehicle_color;

    return 1;

case DRIVER_DATA_WANTS_SUGGESTION :

    *long_value = 1;

    return 1;

case DRIVER_DATA_DESIRED_ACCELERATION : {                                /* START
ACCELERATION FUNCTION */

    double net_distance      = lead_vehicle_distance - lead_vehicle_length;    /* Net
distance */

    double lead_vehicle_speed = current_speed - lead_vehicle_speed_difference; /*
Lead vehicle speed */

    double desired_distance = 2; /* this is in meter. times 1 s = 2.0*/

    /* Changing this value will change the distance from the lead vehicle where the
autonomous car will emergency brake. Make sure this is less than desired_distance. */

    double emergency_stop_distance = 1;

    long regular_cat = current_category;

    long lead_cat = lead_vehicle_category;

```

```

if (regular_cat == lead_cat) {

    if (lead_vehicle_speed_difference > 0){

        /* Faster than the leading vehicle */

        if (lead_vehicle_speed > 0) {

            /* Not behind standstill vehicle (lead_vehicle_speed > 0)*/

            if (net_distance > desired_distance) {

                /* slow down to leading vehicle's speed with 1 s time gap */

                desired_acceleration = - lead_vehicle_speed_difference

                    * lead_vehicle_speed_difference

                    / (net_distance - desired_distance)

                    / 2.0;

            }

        else {

            /* try to increase distance */

            desired_acceleration = - lead_vehicle_speed_difference - 1.0;

            if (net_distance < emergency_stop_distance) {

                desired_acceleration = - lead_vehicle_speed_difference

                    * lead_vehicle_speed_difference

                    / (emergency_stop_distance - net_distance)

                    / 2.0;          /* emergency braking */

            }

        }

    }

}

```

```

else {

    /* leading vehicle is standing still (lead_vehicle_speed = 0)*/

    if (net_distance < emergency_stop_distance) {

        desired_acceleration = - lead_vehicle_speed_difference

            * lead_vehicle_speed_difference

            / (emergency_stop_distance - net_distance)

            / 2.0;          /* emergency braking */

    }

    else {

        /* brake to standstill in 1.0 m distance */

        desired_acceleration = - lead_vehicle_speed_difference

            * lead_vehicle_speed_difference

            / (net_distance - emergency_stop_distance)

            / 2.0;

    }

}

/* -----if (lead_vehicle_speed_difference <= 0)----- */

/* Slower than the leading vehicle */

else {

    /* accelerate to min of leading vehicle's speed and own desired speed */

    /* vehicle is far from leading vehicle: try to decrease distance */

    if (net_distance > desired_distance) {

```

```

        desired_acceleration = lead_vehicle_speed_difference
            * lead_vehicle_speed_difference
            / (net_distance - desired_distance)
            / 2.0;
    }
else {
    /* vehicle is within desired distance of leading vehicle: try to decrease distance
*/
    desired_acceleration = lead_vehicle_speed_difference + 1.0;
    /* vehicle is very close to leading vehicle: try to increase distance */
    if (net_distance < emergency_stop_distance) {
        desired_acceleration = - lead_vehicle_speed_difference
            * lead_vehicle_speed_difference
            / (emergency_stop_distance - net_distance)
            / 2.0;          /* emergency braking */
    }
}

*double_value = desired_acceleration;
}
else {
    *double_value = desired_acceleration;
}

```

```

    return 1;

}

case DRIVER_DATA_DESIRED_LANE_ANGLE :

    *double_value = desired_lane_angle;

    return 1;

case DRIVER_DATA_ACTIVE_LANE_CHANGE :

    *long_value = active_lane_change;

    return 1;

case DRIVER_DATA_REL_TARGET_LANE :

    *long_value = rel_target_lane;

    return 1;

case DRIVER_DATA_SIMPLE_LANECHANGE :

    *long_value = 1;

    return 1;

default :

    return 0;

}

}

/*=====

=====*/

DRIVERMODEL_API int DriverModelExecuteCommand (long number)

{

    /* Executes the command <number> if that is available in the driver */

```

```

/* module. Return value is 1 on success, otherwise 0. */
switch (number) {

    case DRIVER_COMMAND_INIT :

        return 1;

    case DRIVER_COMMAND_CREATE_DRIVER :

        return 1;

    case DRIVER_COMMAND_KILL_DRIVER :

        return 1;

    case DRIVER_COMMAND_MOVE_DRIVER :

        return 1;

    default :

        return 0;

}
}

```

#### resource.rc file

```

#define IDS_STRING1    1

#define IDS_STRING2    2

#define IDS_STRING3    3

STRINGTABLE

{

    IDS_STRING1 "DriverModelSetValue"

    IDS_STRING2 "DriverModelGetValue"

    IDS_STRING3 "DriverModelExecuteCommand"

```

}



## Appendix C

### Code for Integration of Fault Tree and Simulation Modeling

'Programmer: Plaban Das, MS Thesis Work

'Last Update: 7-12-2017

Imports VISSIMLIB

Module Module1

Sub Main()

' Declaration of Variables

Dim Vissim As Object

Dim veh As VISSIMLIB.IVehicle

Dim simend As Integer

Dim vehNo As Integer

' Distance measured in Vissim in meters

Dim total\_dis As Double = 0

Dim over\_single\_sec As Double = 0

Dim j As Integer = 0 : Dim aa As Integer = 0 : Dim bb As Integer = 1 ' : Dim ad As

Integer = 0

Dim total\_distance\_traveled(1, j) As Double : Dim multiplier As Integer = 100 :

Dim crash\_start As Integer = 0

'Dim comp As Integer = 2

' Results from fault tree and convert miles value to meters

```
Dim mile_per_inci As Double = (1000000 / 158) ' Its original value is = 1000000/  
158 = 6330
```

```
Dim conversion_factor As Double = 1609.34 ' Conversion factor from mile to  
meter, as values in vissim are in meters
```

```
Dim response_time As Double = 180 ' Its original value is = 3 mins = 180 sec
```

```
Dim pre_value As Double
```

```
Dim bool As Boolean = False
```

```
Dim random_veh As Integer
```

```
Dim target_veh As Integer
```

```
'Load Vissim file with 32 bit version
```

```
Vissim = CreateObject("Vissim.Vissim-32.700")
```

```
'Load Vissim File from Drive Desired Location
```

```
Vissim.Loadnet("C:\Users\dasp6\Downloads\Research_Autonomous Car_Summer  
2015_Thesis\Autonomous VISSIM\VISSIM Models\Vissim Model_I476\i476  
network.inpx")
```

```
simend = Vissim.Simulation.AttValue("SimPeriod")
```

```
MsgBox(simend)
```

```
For i = 1 To simend
```

```
'Run simulation single step
```

```
Vissim.Simulation.RunSingleStep
```

```
For Each veh In Vissim.Net.Vehicles
```

```
'Search all vehicle
```

```

If veh.AttValue("VehType") = 500 Then

    'Look for autonomous vehicle, they have vehicle type = 400

    vehNo = veh.AttValue("NO")

    total_dis = veh.AttValue("DistTravTotal")

    If aa < bb Then

        total_distance_traveled(0, j) = vehNo

        total_distance_traveled(1, j) = total_dis

        aa = 2

    End If

    For jj = 0 To ((total_distance_traveled.Length / 2) - 1)

        If total_distance_traveled(0, jj) = vehNo Then

            total_distance_traveled(1, jj) = total_dis

            Exit For

        ElseIf (total_distance_traveled(0, jj) <> vehNo) Then

            If (jj < ((total_distance_traveled.Length / 2) - 1)) Then

                GoTo Line1

            ElseIf (jj = ((total_distance_traveled.Length / 2) - 1)) Then

                j = j + 1

                ReDim Preserve total_distance_traveled(1, j)

                total_distance_traveled(0, j) = vehNo

                total_distance_traveled(1, j) = total_dis

            End If

```

```

End If

Line1:      Next

            bool = True

            End If

        Next

If bool = True Then

    over_single_sec = 0

    For ii = 0 To ((total_distance_traveled.Length / 2) - 1)

        over_single_sec = over_single_sec + total_distance_traveled(1, ii)

    Next

End If

'Here is for the crash conditions

If over_single_sec > (multiplier * mile_per_inci * conversion_factor) Then

    'Now total distance is higher than the crash distance

    'Cond=====

    Dim all_veh_count(0) As Integer

    Dim pp As Integer = 0

    For Each veh In Vissim.Net.Vehicles

        If veh.AttValue("VehType") = 500 Then

            If pp = 0 Then

                all_veh_count(pp) = veh.AttValue("NO")

```

```

        GoTo Line2

    End If

    ReDim Preserve all_veh_count(pp)

    all_veh_count(pp) = veh.AttValue("NO")

Line2:        pp = pp + 1

    End If

Next

'Select a vehicle randomly to cause a crash

random_veh = CInt(Int(((all_veh_count.Length) * Rnd()) + 0))

If random_veh = all_veh_count.Length Then

    random_veh = random_veh - 1

End If

target_veh = all_veh_count(random_veh)

place_holder = all_veh_count

'Show the vehicle no to visualize

MsgBox("Crashed Vehicle No:" & target_veh)

multiplier = multiplier + 1

crash_start = i

End If

'Stop the vehicle till response team appear at the crash scene

If (crash_start <> 0) And (i <= (crash_start + response_time)) Then

    If (crash_start = i) Then

```

```

        pre_value =
Vissim.Net.Vehicles.ItemByKey(target_veh).AttValue("DesSpeed")

    End If

    Vissim.Net.Vehicles.ItemByKey(target_veh).AttValue("DesSpeed") = 0
    Vissim.Net.Vehicles.ItemByKey(target_veh).AttValue("Speed") = 0

    ElseIf (crash_start <> 0) And (i = (crash_start + response_time + 1)) Then
        'Response team appeared

        Vissim.Net.Vehicles.ItemByKey(target_veh).AttValue("DesSpeed") =
pre_value

        Vissim.Net.Vehicles.RemoveVehicle(target_veh)

    End If

Next

MsgBox("End")

Vissim = Nothing

End Sub

End
Module

```

## Appendix D

### Survey Calculation

Table 18

*Responses of the question asking failure probability of LIDAR*

Parti- cipants	Set of Options (failure probability ranges) in the question				
	A: < 1.00	B: 1.01 to 3.00	C: 3.01 to 6.00	D: 6.01 to 10.00	E: > 10.00
1	0	0	5	0	0
2	0	0	5	0	0
3	0	0	5	0	0
4	0	0	0	0	5
5	0	0	0	0	5

Number of experts,  $m = 5$

Number of options,  $n = 5$

Now,  $R = \sum_{i=1}^n (R_i - \bar{R})^2 = 200$ , where for each option,  $R_i$  is the sum of the rating participants  $j$  provides to a specific option:  $R_i = \sum_{j=1}^m r_{ij}$  and  $\bar{R}$  is the mean of the  $R_i$ .

$$\text{Kendall's } W = \frac{12 \times R}{m^2 \times (n^3 - n)} = 0.8$$

Table 19

*Responses of the question asking failure probability of camera*

Parti- cipants	Set of Options (failure probability ranges) in the question				
	A: < 1.00	B: 1.01 to 3.00	C: 3.01 to 6.00	D: 6.01 to 10.00	E: > 10.00
1	0	5	0	0	0
2	0	0	5	0	0
3	0	0	0	5	0
4	0	0	0	0	5
5	0	0	0	0	5

As we mentioned before,  $m = 5$ , and  $n = 5$ ,

$$\text{Now, } R = \sum_{i=1}^n (R_i - \bar{R})^2 = 50$$

$$\text{Kendall's } W = \frac{12 \times R}{m^2 \times (n^3 - n)} = 0.2$$



## Appendix E

### Travel Time Data for Travel Time Measurement Segment 1

Table 20

*Travel time data for travel time measurement segment 1*

Random Seed #	Time Intervals	Travel Time (in seconds)				
		AV 0%	AV 10%	AV 25%	AV 50%	AV 90%
5	600-1500	289.32	266.65	237.11	209.74	157.92
	1500-2400	300.07	282.26	259.10	222.87	164.54
	2400-3300	304.58	271.18	244.13	218.49	153.88
	3300-4200	333.91	307.62	266.10	242.97	179.89
10	600-1500	304.57	283.41	255.42	230.15	158.45
	1500-2400	322.29	299.28	294.55	249.50	177.33
	2400-3300	319.85	287.13	267.97	243.23	171.97
	3300-4200	351.96	329.81	316.54	271.81	195.19
15	600-1500	316.30	279.88	253.16	194.40	139.46
	1500-2400	335.71	303.78	295.28	231.00	166.20
	2400-3300	333.11	287.08	257.53	226.92	153.18
	3300-4200	360.21	334.32	319.28	240.06	177.23
20	600-1500	295.44	283.38	268.69	227.83	145.82
	1500-2400	321.49	311.38	291.27	251.30	160.32
	2400-3300	328.47	299.26	270.88	236.88	151.47
	3300-4200	342.90	333.66	308.23	266.95	194.63
25	600-1500	322.43	292.65	262.04	232.97	126.91
	1500-2400	343.74	312.69	304.28	254.04	164.57
	2400-3300	355.64	295.14	298.27	249.88	147.67
	3300-4200	387.65	344.59	315.59	277.64	192.37
30	600-1500	337.17	296.44	229.94	194.06	163.42
	1500-2400	349.18	307.72	244.90	204.38	182.59
	2400-3300	343.59	297.06	232.36	208.50	171.23
	3300-4200	386.76	331.37	274.92	237.34	191.15
35	600-1500	311.58	276.66	256.72	223.36	123.92
	1500-2400	328.20	303.24	291.65	265.91	164.36
	2400-3300	334.11	288.00	277.25	231.32	141.48
	3300-4200	367.88	318.80	309.53	269.46	183.11

Table 20 (continued)

Random Seed #	Time Intervals	Travel Time (in seconds)				
		AV 0%	AV 10%	AV 25%	AV 50%	AV 90%
40	600-1500	293.80	287.84	268.59	234.22	146.37
	1500-2400	307.55	311.62	292.64	264.60	165.73
	2400-3300	314.11	298.85	271.67	255.23	162.30
	3300-4200	331.61	336.49	317.71	268.70	185.92
45	600-1500	323.42	285.01	265.29	235.34	166.89
	1500-2400	346.82	292.69	304.80	257.62	183.85
	2400-3300	334.58	290.45	295.39	267.06	172.48
	3300-4200	391.14	313.02	326.77	268.77	195.85
50	600-1500	328.50	296.95	262.71	226.23	135.08
	1500-2400	354.56	304.14	298.85	248.91	143.72
	2400-3300	368.89	288.94	263.83	249.60	137.09
	3300-4200	397.45	349.28	306.85	261.70	150.56
55	600-1500	296.95	277.67	230.40	198.18	159.90
	1500-2400	321.51	297.83	264.25	239.20	175.99
	2400-3300	319.67	282.62	239.85	231.53	147.96
	3300-4200	347.95	321.27	271.24	243.94	197.84
Average	600-1500	310.9	284.2	253.6	218.8	147.6
	1500-2400	330.1	302.4	285.6	244.5	168.1
	2400-3300	332.4	289.6	265.4	238.1	155.5
	3300-4200	363.6	329.1	303.0	259.0	185.8

## Appendix F

### Conflict Analysis for Different Autonomous Vehicle Penetrations

Table 21

*Conflict analysis for different autonomous vehicle penetrations (TTC= 1.5 and PET = 4.0)*

AV Percentages	Random Seed #	Crossing Conflicts	Lane Change Conflicts	Rear End Conflicts	Sub-total
AV 0%	5	6	130180	2868	133054
	10	9	129848	3056	132913
	15	20	143679	2949	146648
	20	24	143567	3066	146657
	25	11	153577	3138	156726
	30	34	133322	3091	136447
	35	27	152268	3269	155564
	40	3	132972	2943	135918
	45	13	137473	3080	140566
	50	7	149728	3188	152923
	55	22	140111	2900	143033
Total		176	1546725	33548	1580449
AV 10%	5	8	110036	2873	112917
	10	10	113734	2927	116671
	15	5	117349	2951	120305
	20	6	127766	2993	130765
	25	11	121691	3042	124744
	30	7	120493	3207	123707
	35	5	127636	3239	130880
	40	10	112945	2990	115945
	45	7	115630	3104	118741
	50	12	134583	3327	137922
	55	8	122351	3052	125411
Total		89	1324214	33705	1358008

Table 21 (continued)

AV Percentages	Random Seed #	Crossing Conflicts	Lane Change Conflicts	Rear End Conflicts	Sub-total
AV 25%	5	5	73083	1812	74900
	10	2	80067	1974	82043
	15	4	70037	1779	71820
	20	3	85475	1819	87297
	25	1	81518	1907	83426
	30	3	78988	1862	80853
	35	3	79628	1985	81616
	40	6	81256	2046	83308
	45	3	74018	1927	75948
	50	7	92025	1996	94028
	55	2	75647	1815	77464
Total		39	871742	20922	892703
AV 50%	5	5	45968	852	46825
	10	1	47984	933	48918
	15	0	45807	776	46583
	20	3	50316	997	51316
	25	5	45602	881	46488
	30	3	48442	937	49382
	35	1	48783	901	49685
	40	1	48079	846	48926
	45	0	44270	771	45041
	50	1	50066	968	51035
	55	3	46927	859	47789
Total		23	522244	9721	531988
AV 90%	5	1	36906	277	37184
	10	0	38017	281	38298
	15	0	36233	296	36529
	20	1	39108	301	39410
	25	0	35920	230	36150
	30	1	38395	321	38717
	35	0	37105	325	37430
	40	0	37622	276	37898
	45	1	36290	256	36547
	50	1	38655	273	38929
	55	2	36270	272	36544
Total		7	410521	3108	413636